A Modern Code for Solving Magneto-hydrodynamic or Hydrodynamic Equations[1]

Shengtai Li

Theoretical Division, Los Alamos National Laboratory

Hui Li

Applied Physics Division, Los Alamos National Laboratory

ABSTRACT

We have developed a modern code to solve the magneto-hydrodynamic (MHD) or hydrodynamic (HD) equations. The code consists of several approaches for solving the MHD (or HD) by high-resolution schemes with finite-volume and finite difference methods. A framework that implements adaptive mesh refinement (AMR) with nested block-structure is developed. Cylindrical and spherical geometries are considered as orthogonal curvilinear grid as well as the Cartesian grid. We also enhance the AMR capabilities by the preserving the conservative quantities and preserving the divergence free constraint for vector-field. The code is fully parallelized with message passing interface (MPI) and a dynamic load balancing scheme is incorporated to improve the parallel efficiency. Our code is designed in a way that existing codes for a single grid can be easily incorporated.

1. Introduction

Numerical simulation for astrophysics phenomena becomes more and more popular and important in the last decades. Many astrophysics problems can be formulated as hydrodynamics (HD) or magneto-hydrodynamics (MHD) system of equations. Therefore, many numerical simulation codes are based on correctly solving these equations.

[1] Los Alamos Report LA-UR-03-8925

There are several codes available: ZEUS (Stone & Norman 1992) from NCSA, FLASH (Fryxell et al. 2000) from flash center of the University of Chicago, BATSRUS (Powell et al. 1999)from university of Michigan, and many others.

We do not try to "re-invent the wheel" here but rather to develop a versatile code that can be used easily, efficiently by application scientists. We admitted our effort might not be as good as those available codes, which still keep on updating and improving. However, we do have some new features that may not be available in other codes.

For testing, comparison and verification, we included several *basic solvers* in our code. The "basic solver" means a complete solver for a nonadaptive single rectangular grid. For a time-dependent PDE problem, a complete basic solver has several components: grid generation, initial conditions, boundary conditions, spatial discretization and time integration. For a fully-discretized solver, the spatial discretization and time integration is often mixed and can be treated as a black-box that outputs the solution at next time level given the input values at the current time level. The Lax-Wendroff type of scheme belongs to such kind of solver. Currently, we have implemented HLLE, HLLC, and Roe's approximate Riemann solvers and Colella's multi-dimensional scheme for second-order method, PPM (available only for HD) for for third-order, and WENO for fifth-order method. Our framework has a capacity that other solvers can be incorporated easily without much recoding.

A real astrophysics problem contains multiple time and length scales that must be resolved simultaneously. That's why we adopt the adaptive mesh refinement approach in our implementation. Our adaptive mesh refinement framework basically inherits all of the features of Berger and Colella's method (Berger & Colella 1989). We also enhanced Berger's AMR by an improved clustering algorithm (Li & Hyman 1998b), by allowing staggered grid variables for vector-field components, by adding the cylindrical and spherical geometry. We implemented our novel approach (Li & Li 2003) for maintaining divergence free condition for AMR on any curvilinear grid. Moreover, Our AMR framework did not start from scratch. It is based on a small version developed by our first author in his Ph.D. study (Li 1998).

As computers evolves, it became evident that the most powerful computers would be massively parallel computers. The AMR's block data structure has a nature to be parallelized easily. Each block (or *patch*) can be assigned to a processor independently. However, it became increasing apparent that applications that used structured AMR techniques could not be efficiently parallelized due to hierarchical data structure that requires more communication between grids on different levels. As a result, various C++class library-based approach emerged which purported to help in the parallelization of AMR applications. The most prominent examples include the CHOMBO (Colella et al. 2003) and DAGH (Mitra et al. 1997) library packages. The recursive feature of the time integration for a hierarchical grid also requires the integration for different levels must be done sequentially. In our framework, we proposed a locked time step method for a class of second order numerical schemes that allows the grids on different refinment level can be integrated independently during a time step. The computation and communication are interlaced to achieve full parallelization.

The outline of our report is as follows. In section 2, we describe the MHD equations for general orthogonal curvilinear grid. In section 3 we describe several basic solver for a single nonadaptive grid. In section 4, we describe our AMR framework. In section 5 we describe our strategy on parallelizing the AMR hierarchical data structure. In section 6 we describe some implementation details on spatial discretization and AMR operations for cylindrical and spherical grids. Several benchmark problems are presented in section 7 to demonstrate and test the efficiency and accuracy of our code.

2. MHD Equations for Orthogonal Coordinates

The equations of ideal MHD system can be formulated as

$$\begin{aligned}
\rho_t + \nabla\cdot(\rho\mathbf{u}) &= 0, && (1)\\
(\rho\mathbf{u})_t + \nabla\cdot\left[\rho\mathbf{u}\mathbf{u}^T + (p)I - \mathbf{B}\mathbf{B}^T\right] &= 0, && (2)\\
e_t + \nabla\cdot\left[(e+p)\mathbf{u} - \mathbf{B}(\mathbf{u}\cdot\mathbf{B})\right] &= 0, && (3)\\
\mathbf{B}_t - \nabla\cdot(\mathbf{u}\mathbf{B}^T - \mathbf{B}\mathbf{u}^T) &= 0, && (4)
\end{aligned}$$

where ρ is density, $\mathbf{u}$ is the velocity, $\mathbf{B}$ is the magnetic field, e is the total energy per unit volume, and p is the total pressure, defined as

$$p = p_{\text{gas}} + \frac{1}{2}\mathbf{B}\cdot\mathbf{B},$$

where p_{gas} is the gas pressure that satisfies the equation of state,

$$p_{\text{gas}} = (\gamma-1)\left(e - \frac{1}{2}\rho\mathbf{u}\cdot\mathbf{u} - \frac{1}{2}\mathbf{B}\cdot\mathbf{B}\right).$$

One external constraint for magnetic field is

$$\nabla\cdot\mathbf{B} = 0,$$

which becomes $B_x =$ constant in one dimension.

When the magnetic field components $B = 0$, the ideal MHD equations reduce to the Euler equations for hydrodynamics. For briefness, we discribe only the MHD equations and their eigen-system in the following. The Euler equations and their corresponding eigen-system can be derived similarly by assigning $B = 0$ in our formulation.

2.1. MHD equations for local coordinate system

In this work, we are primarily interested in three coordinate systems: Cartesian coordinates for which

$$(x_1, x_2, x_3) = (x, y, z), \qquad (h_1, h_2, h_3) = (1, 1, 1),$$

cylindrical coordinates for which

$$(x_1, x_2, x_3) = (r, z, \phi), \qquad (h_1, h_2, h_3) = (1, 1, r),$$

and spherical polar coordinates for which

$$(x_1, x_2, x_3) = (r, \theta, \phi), \qquad (h_1, h_2, h_3) = (1, r, r\sin\theta),$$

where h_i are the metric scale factors. If not stated otherwise, x_3 is always the ignorable coordinates for a reduced 2-D problem,

The conservative form of the MHD equations are a slight different for different geometries, because the divergence operator $\nabla\cdot$, gradient operator ∇, and curl operation $\nabla\times$ have different forms in different coordinates. Basically for the gradient of a scalar function, we have

$$\nabla f = \left(\frac{1}{h_1}\frac{\partial f}{\partial x_1}, \frac{1}{h_2}\frac{\partial f}{\partial x_2}, \frac{1}{h_3}\frac{\partial f}{\partial x_3}\right);$$

for the divergence of a vector $A = (a_1, a_2, a_3)$, we have

$$\nabla \cdot A = \frac{1}{h_1 h_2 h_3}\left(\frac{\partial}{\partial x_1}(h_2 h_3 a_1) + \frac{\partial}{\partial x_2}(h_1 h_3 a_2) + \frac{\partial}{\partial x_3}(h_1 h_2 a_3)\right);$$

for the curl of the vector A, we have

$$\begin{aligned}\nabla \times A \quad = \quad & \left\{\frac{1}{h_2 h_3}\left(\frac{\partial}{\partial x_2}(h_3 a_3) - \frac{\partial}{\partial x_3}(h_2 a_2)\right),\right.\\ & \frac{1}{h_1 h_3}\left(\frac{\partial}{\partial x_3}(h_1 a_1) - \frac{\partial}{\partial x_1}(h_3 a_3)\right),\\ & \left.\frac{1}{h_1 h_2}\left(\frac{\partial}{\partial x_1}(h_2 a_2) - \frac{\partial}{\partial x_2}(h_1 a_1)\right)\right\}.\end{aligned}$$

For the gradient of a vector $A = (a_1, a_2, a_3)$, we have

$$\nabla A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix},$$

where a_{ij} is

$$a_{ij} = \begin{cases} \frac{1}{h} i \frac{\partial a_j}{\partial x_i} - \frac{a_i}{h_i h_j} \frac{\partial h_i}{\partial x_j}, & \text{if } i \neq j \\ \frac{1}{h} i \frac{\partial a_i}{\partial x_i} + \frac{a_l}{h_i h_l} \frac{\partial h_i}{\partial x_l} + \frac{a_k}{h_i h_k} \frac{\partial h_i}{\partial x_k}, & \text{if } i = j, (l,k) \neq i \end{cases}$$

Insert these equations into MHD equations and regroup them, we can obtain the MHD equations in conservative form:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{1}{h_1 h_2 h_3} \left(\frac{\partial}{\partial x_1}(h_2 h_3 \mathbf{F}) + \frac{\partial}{\partial x_2}(h_1 h_3 \mathbf{G}) + \frac{\partial}{\partial x_3}(h_1 h_2 \mathbf{H}) \right) = \mathbf{S}, \tag{5}$$

where

$$\mathbf{q} = (\rho, \rho v_1, \rho v_2, \rho v_3, B_1, B_2, B_3, E)^t,$$

and the flux functions are

$$\mathbf{F} = \begin{pmatrix} \rho v_1 \\ \rho v_1^2 - B_1^2 + p^* \\ \rho v_1 v_2 - B_1 B_2 \\ \rho v_1 v_3 - B_1 B_3 \\ 0 \\ \Omega_3 \\ -\Omega_2 \\ (E + p^*) v_1 - B_1 (\mathbf{B} \cdot \mathbf{v}) \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \rho v_2 \\ \rho v_2 v_1 - B_2 B_1 \\ \rho v_2^2 - B_2^2 + p^* \\ \rho v_2 v_3 - B_2 B_3 \\ -\Omega_3 \\ 0 \\ \Omega_1 \\ (E + p^*) v_2 - B_2 (\mathbf{B} \cdot \mathbf{v}) \end{pmatrix},$$

$$\mathbf{H} = \begin{pmatrix} \rho v_3 \\ \rho v_3 v_1 - B_3 B_1 \\ \rho v_3 v_2 - B_3 B_2 \\ \rho v_3^2 - B_3^2 + p^* \\ \Omega_2 \\ -\Omega_1 \\ 0 \\ (E + p^*) v_3 - B_3 (\mathbf{B} \cdot \mathbf{v}) \end{pmatrix},$$

and the source terms are

$$\mathbf{S} = \left(\begin{array}{c} 0 \\ \frac{B_1B_2-\rho v_1v_2}{h_1h_2}\frac{\partial h_1}{\partial x_2} + \frac{B_1B_3-\rho v_1v_3}{h_1h_3}\frac{\partial h_1}{\partial x_3} + \frac{\rho v_2^2-B_2^2}{h_1h_2}\frac{\partial h_2}{\partial x_1} + \frac{\rho v_3^2-B_3^2}{h_1h_3}\frac{\partial h_3}{\partial x_1} + \frac{p^*}{h_1h_2h_3}\frac{\partial (h_2h_3)}{\partial x_1} \\ \frac{B_2B_1-\rho v_2v_1}{h_1h_2}\frac{\partial h_2}{\partial x_1} + \frac{B_2B_3-\rho v_2v_3}{h_2h_3}\frac{\partial h_2}{\partial x_3} + \frac{\rho v_1^2-B_1^2}{h_1h_2}\frac{\partial h_1}{\partial x_2} + \frac{\rho v_3^2-B_3^2}{h_2h_3}\frac{\partial h_3}{\partial x_2} + \frac{p^*}{h_1h_2h_3}\frac{\partial (h_1h_3)}{\partial x_2} \\ \frac{B_3B_2-\rho v_3v_2}{h_3h_2}\frac{\partial h_3}{\partial x_2} + \frac{B_1B_3-\rho v_1v_3}{h_1h_3}\frac{\partial h_3}{\partial x_1} + \frac{\rho v_1^2-B_1^2}{h_1h_3}\frac{\partial h_1}{\partial x_3} + \frac{\rho v_2^2-B_2^2}{h_2h_3}\frac{\partial h_2}{\partial x_3} + \frac{p^*}{h_1h_2h_3}\frac{\partial (h_1h_2)}{\partial x_3} \\ \frac{\Omega_2}{h_1h_3}\frac{\partial h_1}{\partial x_3} - \frac{\Omega_3}{h_1h_2}\frac{\partial h_1}{\partial x_2} \\ \frac{\Omega_3}{h_1h_2}\frac{\partial h_2}{\partial x_1} - \frac{\Omega_1}{h_3h_2}\frac{\partial h_2}{\partial x_3} \\ \frac{\Omega_1}{h_2h_3}\frac{\partial h_3}{\partial x_2} - \frac{\Omega_2}{h_1h_3}\frac{\partial h_3}{\partial x_1} \\ 0 \end{array} \right)$$

where

$$p^* = p + \frac{1}{2}\mathbf{B}\cdot\mathbf{B}$$

is the total pressure,

$$\Omega_1 = v_2B_3 - v_3B_2, \quad \Omega_2 = v_3B_1 - v_1B_3, \quad \Omega_3 = v_1B_2 - v_2B_1$$

are the electromotive force (EMF, defined via $\Omega = \mathbf{v} \times \mathbf{B}$), and

$$E = \frac{1}{2}\rho\mathbf{v}\cdot\mathbf{v} + \frac{p}{\gamma - 1} + \frac{1}{2}\mathbf{B}\cdot\mathbf{B}$$

is the total energy. Note that for cylindrical and spherical geometry, all h_i are known and the above equations can be much reduced.

The 3-D MHD equations can be reduced to 2-D or even 1-D problems if symmetry of the variables is used. For examples, for cylindrical coordinates, $h_1 = h_2 = 1$ and $h_3 = r$, we can reduce it to (r, ϕ) (or (r, z)) problem if all of the variables are of functions of (r, ϕ) (or (r, z)). It can be further reduced to 1-D problem of r if all of the variables are of functions of r only. For 3-D spherical coordinates, 3-D MHD equations can be reduced to 2-D problem of (r, θ) if radially symmetry is used. It is hard to reduce it to a 1-D problem because $h_3 = r\sin\theta$ appears in both the first and second flux terms, especially for MHD where the magnetic field $\mathbf{B}$ that satisfies the divergence-free condition cannot be a function only of r. The problem reduction is very valuable in testing and validating numerical codes for multidimensional problem. A highly accurate solution to the 1-D or 2-D problem can be computed on a fined grid and used to test solutions computed with fully 3-D solver. This can also be used to detect where the numerical method suffers from grid-orientation effects that lead to the results being better resolved in some directions than in others.

For a real application, the ideal MHD equations may subject to some source terms, such as the external gravity or self-gravitation. For the gravitational MHD systems, the source term is added to the momentum and energy equations. The others remain the same.

2.2. Eigenvalues and eigen-vectors

From the flux functions, we can obtain the Jacobian matrices

$$A_x(\mathbf{q}) = \frac{\partial \mathbf{F}}{\partial \mathbf{q}}, \quad A_y(\mathbf{q}) = \frac{\partial \mathbf{G}}{\partial \mathbf{q}}, \quad A_z(\mathbf{q}) = \frac{\partial \mathbf{H}}{\partial \mathbf{q}},$$

It is observed that G and H can be obtained from properly index permuting. Therefore, A_y and A_z have similar structure as A_x. The eigenvalues and and eigenvectors for A_x have been extensively studied by many authors (Brio & Wu 1988; Ryu & Jones 1995; Roe & Balsara 1996). There are two set of eigenvectors for the eigen-system of A_x. One is a direct extension of the one-dimensional system and the other is based on the modification proposed by Powell et al (Powell et al. 1999). In the direct extension, the 5th component is ignored and the rest is identical to the one-dimensional flux where B_1 is a constant. The eigenvalues for the 7 × 7 systems are

$$\lambda_{1,7} = v_1 \pm c_f, \quad \lambda_{2,6} = v_1 \pm c_a, \quad \lambda_{3,5} = v_1 \pm c_s, \quad \lambda_4 = v_1,$$

where

$$c_a = \sqrt{\frac{B_1^2}{\rho}}$$

is the speed of Alfven waves, and

$$c_f = \left[\frac{1}{2}\left(a^2 + \frac{B^2}{\rho} + \sqrt{\left(a^2 + \frac{B^2}{\rho}\right)^2 - 4a^2 c_a^2}\right)\right]^{\frac{1}{2}},$$

$$c_s = \left[\frac{1}{2}\left(a^2 + \frac{B^2}{\rho} - \sqrt{\left(a^2 + \frac{B^2}{\rho}\right)^2 - 4a^2 c_a^2}\right)\right]^{\frac{1}{2}},$$

are the speeds of fast and slow waves respectively, and a is the speed acoustic wave given by

$$a = \sqrt{\frac{\gamma p}{\rho}}.$$

In the 8 × 8 eigen-system of Powell, the corresponding eigenvalues are

$$\lambda_{1,8} = v_1 \pm c_f, \quad \lambda_{2,7} = v_1 \pm c_a, \quad \lambda_{3,6} = v_1 \pm c_s, \quad \lambda_{4,5} = v_1,$$

The corresponding eigenvectors have been given by many authors. We here adopted the one from Powell et al.(Powell et al. 1999) for the 8-wave eigensystem.

For a dimensional splitting integration scheme, the CFL conditions along each row in three directions can be calculated with (assume the same time step is used)

$$\mathrm{CFL} = dt \cdot \min\left[\frac{|v_1| + c_f}{h_1 dx_1}, \frac{|v_2| + c_f}{h_2 dx_2}, \frac{|v_3| + c_f}{h_3 dx_3}\right],$$

where the minimum is over all of the cells. For a nonsplitting multidimensional scheme, the CFL condition is

$$\mathrm{CFL} = dt \cdot \min\left|\frac{|v_1| + c_f}{h_1 dx_1}, \frac{|v_2| + c_f}{h_2 dx_2}, \frac{|v_3| + c_f}{h_3 dx_3}\right|_2.$$

It is noted that the nonsplitting multi-dimensional scheme may not be as efficient as dimensional splitting schemes due to the restriction on the time step. It is also obvious that when h_2 or h_3 is small, the CFL number becomes potentially large which restricts the time step to be very small. This is one of the disadvantages to use the cylindrical or spherical coordinates instead of the Cartesian grid.

Another difficulty of using cylindrical or spherical coordinates is that it has potential singularity at $r = 0$, and the boundary condition there is also difficult to specify. For cell-centered variables, the singularity problem can be avoided, But for the face-centered variables, such as the magnetic field components, we have to design a scheme to solve or bypass the problem.

3. Basic Solver

In this section, we describe the basic solvers for a single Cartesian grid. These solvers can also be applied to cylindrical or spherical geometry with a slight modification. For verification and numerical test, we have included several solvers in our code. We will describe them one by one.

3.1. Riemann solver

The Godunov method and its high-order extensions require the solution of the Riemann problem. In practical computation this is solved billions of times, making the Riemann problem solution process the single most demanding task in the numerical method. Although the central upwinding scheme, described in (Kurganov et al. 2000) is called Riemann solver free scheme, it is equivalent to HLL Riemann solver (Harten et al. 1983; Einfeldt 1998; Einfeldt et al. 1991).

The exact Riemann solvers requires an iterative procedure to solve a nonlinear problem and the associated computational effort may not always be justified. This effort may increase dramatically by the complexity of the equations of state or the complexity of the particular system of equations being solved. For MHD, the exact Riemann solver is far more costly than the case of HD. In this subsection, we present several approximate Riemann solvers that do not need an iterative process.

3.1.1. HLL or HLLE Riemann solver

The first solver we used is the central upwinding scheme, proposed by Kurganov et al.(Kurganov et al. 2000). This solver is fast and does not need the characteristic decomposition. Although it is called Riemann solver free method, we later found that it was equivalent to HLL Riemann solver (see (Harten et al. 1983; Einfeldt 1998; Einfeldt et al. 1991)) except that the acoustic signal speed is computed differently. The semi-discrete central-upwinding scheme (see (Kurganov et al. 2000)) can be written in the following conservation form,

$$\frac{d}{dt}u_{i,j} = \frac{F_{i+\frac{1}{2},j}(t) - F_{i-\frac{1}{2},j}}{\Delta x} + \frac{G_{i,j+\frac{1}{2}}(t) - G_{i,j-\frac{1}{2}}}{\Delta y},$$

where the numerical fluxes $F_{i+\frac{1}{2},j}$ are given by

$$F_{i+\frac{1}{2},j} = \frac{a^R_{i+\frac{1}{2},j} f(u^R_{i+\frac{1}{2},j}) - a^L_{i+\frac{1}{2},j} f(u^R_{i+\frac{1}{2},j})}{a^R_{i+\frac{1}{2},j} - a^L_{i+\frac{1}{2},j}} + \frac{a^R_{i+\frac{1}{2},j} a^L_{i+\frac{1}{2},j}}{a^R_{i+\frac{1}{2},j} - a^L_{i+\frac{1}{2},j}} \left[u^R_{i+\frac{1}{2},j} - u^L_{i+\frac{1}{2},j} \right],$$

The numerical fluxes $G_{i,j+\frac{1}{2}}$ can be given similarly. Here $u^R_{i+\frac{1}{2},j}$ and $u^L_{i+\frac{1}{2},j}$ stand for the corresponding right and left interface values at $x = x_{j+\frac{1}{2}}$. These values can be calculated by limited reconstruction of a non-oscillatory polynomial for u at $x = x_{j+\frac{1}{2}}$. The one-sided local speeds $a^*_{i+\frac{1}{2},j}$ are determined by

$$\begin{aligned} a^R_{i+\frac{1}{2},j} &= \max\left\{ \lambda_N \left(\frac{\partial f}{\partial u}(u^R_{i+\frac{1}{2},j}) \right), \lambda_N \left(\frac{\partial f}{\partial u}(u^L_{i+\frac{1}{2},j}) \right), 0 \right\}, \\ a^L_{i+\frac{1}{2},j} &= \min\left\{ \lambda_1 \left(\frac{\partial f}{\partial u}(u^R_{i+\frac{1}{2},j}) \right), \lambda_1 \left(\frac{\partial f}{\partial u}(u^L_{i+\frac{1}{2},j}) \right), 0 \right\}, \end{aligned}$$

with $\lambda_1 < ... < \lambda_N$ being the N eigenvalues of the Jacobian $\frac{\partial f}{\partial u}$.

It has been proved in (Einfeldt 1998; Einfeldt et al. 1991) that this method can preserve the positivity, which is important when the kinetic or magnetic energy dominants the total energy, for example, in a low-beta plasma ($\beta = 2\mu_0 P/\mathbf{B}^2$). The pressure can be kept positive during time integration. Einfeldt also suggests that one-sided local speeds are determined by

$$\begin{aligned} a^R_{i+\frac{1}{2},j} &= \max\left\{\lambda_N\left(\frac{\partial f}{\partial u}(u^R_{i+\frac{1}{2},j})\right), \lambda_N\left(\frac{\partial f}{\partial u}(u^a_{i+\frac{1}{2},j})\right), 0\right\}, \\ a^L_{i+\frac{1}{2},j} &= \min\left\{\lambda_1\left(\frac{\partial f}{\partial u}(u^a_{i+\frac{1}{2},j})\right), \lambda_1\left(\frac{\partial f}{\partial u}(u^L_{i+\frac{1}{2},j})\right), 0\right\}, \end{aligned}$$

where u^a is the Roe's average states between u^R and u^L.

The HLL Riemann solver has strong diffusion on the rarefaction waves and in the contact field. Several anti-diffusion improvements have been proposed. HLLC and HLLEM are two of them.

3.1.2. HLLC Riemann solver

The HLLC, proposed by Toro, et al. (Toro et al. 1994), is a modification of the HLL scheme, whereby the missing contact and shear waves are restored. It replaces the constant intermediate state u^* between the fastest and slowest waves with two intermediate state u^{*R} and u^{*L}. The middle wave of speed a^* is inserted between slowest and fastest signal speed a^R and a^L. For hydrodynamics problem, we adopted Batten et al. (Batten et al. 1997) approach to calculate the middle speed a^*.

Although MHD has more waves than HD, the HLLC can still be used after modification. Following the derivation of Toro et al. (Toro et al. 1994), we derived a version of HLLC for MHD. By applying Rankine-Hugoniot conditions across each of the waves of speeds a^L, a^*, a^R, and assumption of the pressure remains constant across the contact wave, we derived the middle wave speed a^* to be

$$a^* = \frac{p_R - p_l + (B_x)^2_L - (B_x)^2_R + \rho_L u_L(a^L - u_L) - \rho_R u_R(a^R - u_R)}{\rho_L(a^L - u_L) - \rho_R(a^R - u_R)},$$

where p is the total pressure that consists of both the thermal and magnetic pressure. Then the left and right middle state u^{*L} and U^{*R} are derived based on the Rankine-Hugoniot conditions.

We found by numerical experiments that this approach did not work well for MHD problem. The pressure can easily become negative. The problems occur when the middle state of magnetic fields exceeds both the left and right states. So we enforce one constant middle state for magnetic field across the slowest and fastest waves. The middle state of the magnetic field can be given by the HLL Riemann solver. The rest of variables are calculated via Rankine-Hugoniot conditions. This approach is reduced to the HD case when the magnetic field components are all zeros. Detail of our discussion and derivation of HLLC solver for MHD is described in (Li & Li 2003).

HLLC can dramatically improve the results of HLL solver. The increase in computational cost is much less than the HLLEM method, which will be described in the following subsection, because HLLC does not need eigen-decompositions.

3.1.3. HLLEM Riemann solver

HLLEM is another approach, proposed in (Einfeldt et al. 1991), to improve the results of the HLL solver for contact wave. An anti-diffusion term is added to the original HLLE flux to reduce the diffusion at the contact wave, which results in

$$F^{\text{HLLEM}} = F^{\text{HLLE}} - \frac{a^R a^L}{a^R - a^L} \sum_p \delta \alpha^p \mathbf{R}^p.$$

where the sum is over all of the relevent p, $\mathbf{R}^p$ are the right eigenvectors of the flux Jacobian evaluated at the intermediate states, α^p are the coefficient of the projection of $u^R - u^L$ onto $\mathbf{R}^p$,

$$u^R - u^L = \sum_p \alpha^p \mathbf{R}^p,$$

and δ is the anti-diffusion coefficient. It is also proposed as a modification to original Roe's approximate Riemann solver to preserve the positivity.

For HD, the anti-diffusion term is added only to the contact wave. For a MHD problem, the anti-diffusion term needs to be added to all but the fast magneto-sonic waves. It is claimed in (Wesenberg 2002) that the MHD-HLLEM solver is the most effiicient solver in terms of computational time versus error among several solvers that have been tested. However, as we have seen, to calculate the anti-diffusion term, partial eigen-decomposition is required. That's why it costs more than the HLLC solver.

3.1.4. Roe's original Riemann solver

We also implemented the Roe's original approximated Riemann solver for our MHD code. Both seven-wave and eight-wave eigen-systems are implemented. The detail of the Roe's solver for MHD are described in (Powell et al. 1999). In order to handle the source term correctly, we used an unsplitting and semi-discretized version of the Roe's method. An entropy fix is also implemented to improve the robustness of the method.

3.1.5. Hybrid Riemann solver

We also combined the HLLE solver and Roe's Riemann solver together in our MHD code. As proposed in (Janhunen 2000), we used the Roe's Riemann solver during the computation but also calculated the HLLE middle state. If any unphysical states (e.g., negative pressure) occur, we instead use the HLLE Riemann solver.

3.2. Data Reconstruction

The above Riemann solvers are proposed for 1-D problem and first order method. To achieve the high order accuracy in space, the second or higher order reconstruction is required. We have included several data reconstructions in our code. It was observed that even if the discretization of the scheme is of second order, high order (higher than 2) data reconstructions have better resolution near discontinuity. That's why we also include several 4th order or 3rd order data reconstruction in our solver. We should mention that for robustness, the data reconstruction should be applied to the primitive variables rather than the conservative variables. The proposed data reconstruction works very well for the second order schemes. However, for other high order methods, the reconstruction in the characteristic space might be needed to avoid the spurious oscillations.

3.2.1. Piecewise linear reconstruction

In this reconstruction, we use piecewise linear profile,

$$u_i(x) = u_i + \frac{x - x_i}{\Delta x}\overline{\Delta u_i},$$

where $\overline{\Delta u_i}$ is a limited slope for cell $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$. There are several slope limiters available. In our solver, we use two types of limiter. The first type has form

$$\overline{\Delta u_i} = \text{minmod}\left(\theta(u_{i+1} - u_i), \frac{1}{2}(u_{i+1} - u_{i-1}), \theta(u_i - u_{i-1})\right),$$

which is a one-parameter family of the *minmod* limiters and $\theta \in [1, 2]$. Note that larger θ corresponds to less dissipation, but still non-oscillatory limiters. For $\theta = 2$, it becomes Woodward limiter. These minmod type limiters are not smooth functions with respect to u. They will raise convergence difficulty for an iterative method in either steady-state solutions or in an implicit time solver.

The other limiter we used is a smooth function of u,

$$\overline{\Delta u_i} = \frac{2(u_{i+1} - u_i)(u_i - u_{i-1}) + \varepsilon}{(u_{i+1} - u_i)^2 + (u_i - u_{i-1})^2 + \varepsilon} \frac{1}{2}(u_{i+1} - u_{i-1}),$$

where ε is a tiny positive constant in case of $u_i = u_{i+1}$ and $u_i = u_{i-1}$. This limiter is smooth and it preserves the monotonicity of the original profile of u. However, the contact discontinuity resolved by this limiter is not as sharp as the minmod limiters with large β value.

3.2.2. Central WENO reconstruction

WENO reconstruction can achieve higher than second order accuracy. There are two types of the WENO reconstruction: upwinding WENO reconstruction, which is usually implemented in characteristic space, and the central WENO reconstruction, which is implemented directly to the component variables.

We have included three implementations for our central WENO reconstruction. The first one, which was proposed in (Levy et al. 2000) uses a five-point stencil to achieve 4th order accuracy. The second one, called compact central WENO which was proposed in (Levy et al. 2002), uses a three-point stencil to achieve 3rd accuracy. The third one is a fully 2-D compact central WENO reconstruction that was proposed in (Levy et al. 2002). The difference between the second and the third one is that the third one included a mix-derivative in the construction polynomial to reflect a fully 2-D interpolation.

3.3. Second order in time

In this subsection, we discuss how to achieve the second order accuracy in time. We will describe three approaches that are implemented in our solver.

3.3.1. MUSCL Hancock approach

MUSCL stands for monotone upstream-centered scheme for conservative laws. It was proposed by van Leer (van Leer 1979). One of key steps for MUSCL approach is high order monotone-preserved data reconstruction, which we described in section 3.2. In this subsection, we describe one of the MUSCL implementation—MUSCL Hancock method. It is implemented in our both dimensional splitting and unsplitting version. It consists of a prediction and a correction step. In the prediction step, we calculated the value $u^{n+\frac{1}{2}}$ at time $t_{n+\frac{1}{2}}$ (here we discribe only the 1-D case)

$$u_i^{n+\frac{1}{2}} = u_i^n \quad + \quad \frac{1}{2}\frac{\Delta t}{\Delta x}\left(F(u_i^n + \frac{1}{2}\overline{\Delta_x u_i^n}) - F(u_i^n - \frac{1}{2}\overline{\Delta_x u_i^n})\right).$$

where $\overline{\Delta_x u_i^n}$ is the limited slope that is obtained from the data reconstruction. In the next step we use $u_i^{n+\frac{1}{2}}$ and $\overline{\Delta_x u_i^n}$ to construct the left and right interface values at time $t_{n+\frac{1}{2}}$. In x direction, we have

$$u_{i+\frac{1}{2}}^L = u_i^{n+\frac{1}{2}} + \frac{1}{2}\overline{\Delta_x u_i^n}, \quad u_{i+\frac{1}{2}}^R = u_{i+1}^{n+\frac{1}{2}} - \frac{1}{2}\overline{\Delta_x u_{i+1}^n}.$$

Finally, a Riemann problem with left and right values is solved by any Riemann solver we discribed in section 3.1..

Since the Hancock prediction does not need the characteristic decomposition, it can be done fast. It is also noted that the limited slopes are reconstructed only once during the whole process. Therefore, the MUSCL-Hancock approach is much more efficient than a general method of lines (MOL) second-order time integration approach. Like the numerical methods of the second order fully-discretized PDE, the MUSCL-Hancock approach also has benefit in parallel and AMR computation because only one data communication is required during a full-step time integration.

3.3.2. Lax-Wendroff type of integration

This approach based on the idea of classical Lax-Wendroff scheme, and it relies on converting all the time derivatives in a temporal Taylor expansion into spatial derivatives by repeated using the PDE and its differentiated versions. The second-order version is quite simple and has been extensively used for solving multi-dimensional problems with dimensional splitting.

For Godunov method, the key idea is to construct a time-centered spatially second order accurate variables or fluxes at the zone interface. Ryu and Jones (Ryu & Jones 1995) extended Harten's second order method for HD to MHD to construct the time-centered fluxes. Balsara (Balsara 1998) extended Colella's method to construct the time-centered variables and then used them in Riemann solvers. Both methods are constructed for 1-D problem. For multi-dimensional problems, it requires dimensional splitting approach.

3.3.3. Multi-stage integration

Mutl-stage integration is often called the method of lines (MOL) approach. The spatial discretization is first used to obtain a semi-discretized ordinary differential equation (ODE) (or differential-algebraic equation (DAE)) system, and then an ODE/DAE solver is used to advance the time to the next level. We have implemented the improved Euler method, second order Runge-Kutta method and third order TVD Runge-Kutta method in our code. All of the methods we used are explicit time integration. The implicit time integration might be necessary for the equations that contain the diffusion or viscosity term. We have developed an implict AMR method (Li et al. 2001) for such problems.

3.4. Higher order method

We have three high than second order methods implemented for HD: third order central WENO scheme, piece-wise parabolic method (PPM), and fifth order WENO finite-difference scheme. Currently only the fifth order WENO scheme is implemented for MHD. When a high order scheme is used, the number of ghost cells may need to increase. For example, the PPMLR from VH1 code uses 6 ghost cells and WENO three ghost cells.

The fifth-order WENO schemes described in Jiang and Shu (Jiang & Shu 1997) combined with third-order TVD Runge-Kutta method is implemented in our code.

3.5. Multi-dimensional problem

The easiest way to extend one-dimensional numerical schemes to more space dimensions is to use *dimensional splitting* approach. A multidimensional problem is simly split into a sequence of one-dimensional problems. This approach has been widely used by many numerical methods including the PPM method. To minimize the splitting error of this approach, the Strang splitting (Strang 1966) is often adopted. For example, for a 2D dimensional splitting method, if L^x_{dt} stands for the integration in x-direction and L^y_{dt} stands for the integration in y-direction, then $L^x_{dt}L^y_{dt}L^y_{dt}L^x_{dt}$ stands for a whole cycle of the integration that contains two time steps,

$$u^{n+2} = L^x_{dt}L^y_{dt}L^y_{dt}L^x_{dt}(u^n). \tag{6}$$

If the 1D operator is second order accurate, the Strang splitting is second order accurate.

Note that the time step in each operator of (6) must be the same to achieve the second order accuracy, which is harder to implement in connection with variable-size time steps.

Another approach we used is the unsplit scheme, which is very much like a MOL approach. The PDE system is first discretized into an ODE system, and then the Runge-Kutta time stepping method is used. The spatial discretization in each direction for a finite-difference method is simply a spatial discretization for a 1-D problem. Finite-difference WENO schemes use this approach. For a finite-volume method, the one-dimensional Riemann solver can be used as a basic tool in the determination of the interface fluxes.

3.6. Source term and multi-dimensional problem

The system of equations for conservation law can have many source terms. The stiff source terms can arise in the reacting flow and combustion problem. The source term can act as an external force to the fluid, such as gravity. The Geometric source terms can be generated when a phyiscal problem in three space dimensions is reduced to a mathematical problem in one or two dimensions by taking advantages of known symmetries in the solution.

Like a multi-dimensional problem, the source term can be solved by a fractional step method. We assume the source term contains no spatial derivatives. Then the Strang splitting can be implemented as

$$u^{n+1} = S(\frac{1}{2}\Delta t)L(\Delta t)S(\frac{1}{2}\Delta t)u^n, \tag{7}$$

where $S(\frac{1}{2}\Delta t)$ is an operator for source term. Clawpack (Leveque 2003) has implemented this approach for source term.

For a fractional step method like (7), we typically need to impose boundary conditions in the hyperbolic step of the procedure. The boundary conditions for the original PDE must be used to determine any boundary conditions needed fro the fractional steps, but the connection between these is often nontrival. A more general procedure for deriving the proper intermediate boundary conditions for a linear hyperbolic equations is discussed in (Leveque 1986).

There are some other potential pitfalls in using a factional-step method to handle source terms in relation to computing a steady-state solution, or a hydrostatic equilibrium ambient field. In the fractional step method, the hydrodynamics and the external gravity force are not very closely coupled. However, the maintaining the hydrostatic equilibrium (HSE)relies on the exact cancellation of the two possibly large term, the pressure gradient and the gravity force, which are calculated in different way. Spurious velocities are generated when the cancellation is not exact. Flash has applied a special treatment to the Riemann solver and the boundary conditions to maintain HSE.

We prefer to use the unsplit version when solving problems with source terms. It is easy to add source term to the semi-discretization system. Then the system of ODEs os solved by a stable ODE solver.

3.7. Constraint transport for magnetic field

We used the staggered mesh technique, which was proposed by Balsara and Spicer (Balsara & Spicer 1999), to preserve the divergence-free of the magnetic field. We define the magnetic field components on the face centers and all the other fluid quantities are still defined at cell centers. For use in step of calculating the fluxes by the Godunov methods, we also define the magnetic field components at the cell centers, say as intermediate variables by averaging the values at the face centers. The detail of the mapping from the face-centered values to the cell-centered values is described in Section 3.8.

The electro-magnetic force $\Omega = -u \times B$ are defined by averaging the fluxes calculated at the grid interfaces: namely

$$\Omega_{i,j} = \frac{1}{2}(f_{y,i+\frac{1}{2},j} + f_{y,i-\frac{1}{2},j}) - \frac{1}{2}(f_{x,i,j+\frac{1}{2}} + f_{x,i,j-\frac{1}{2}}).$$

Then the magnetic field components are updates by Yee's method,

$$\begin{aligned} B^{n+1}_{x,i-\frac{1}{2},j} &= B^{n}_{x,i-\frac{1}{2},j} - \frac{\Delta t}{\Delta y}(\Omega_{i,j} - \Omega_{i,j-1}), \\ B^{n+1}_{y,i,j-\frac{1}{2}} &= B^{n}_{y,i,j-\frac{1}{2}} + \frac{\Delta t}{\Delta x}(\Omega_{i,j} - \Omega_{i-1,j}). \end{aligned} \tag{8}$$

It is easy to verify that the divergence free is preserved exactly by the above updating if the divergence free condition is satisfied at step n.

Dai and Woodward (Dai & Woodward 1998) proposed another staggered mesh approach which can be taken as a post-processing procedure for the Godunov method. In this approach, all the variables at the cell-center (including those intermediate magnetic components) are advanced in time first. Then spatial and temporal interpolation is used to obtain the cell corner centered magnetic field and velocity field. The electric field at the cell corner is then calculated and Yee's method (8) is used to updated the magnetic field at the face center.

Toth (Tóth 2000) proposed a constrained transport/central-difference (CT/CD) method which works directly on the cell-centered values of the magnetic field. Although it is equivalent to the staggered grid method (Balsara & Spicer 1999) for a single mesh, it does not work under AMR framework. That is to say, it is difficult (or impossible?) to define an prolongation reconstruction to preserve the divergence free condition when the mesh is adapted.

3.8. From face-centered value to cell-centered value

The implementation of Godunov type Riemann solver for MHD requires all of the variables stay at the cell-centers as finite-volume average. Given the field components located at the face center, we can calculate the cell-average value of the field components located at the cell-center. We illustrate it by a 2-D problem.

Assume linear profile is used in reconstruction of face values on the coarse face. The coarse cell is $[x_1, x_2] \times [y_1, y_2]$, which may not be Cartesian grid. The linear profile on upper and lower face is

$$B^{\pm}_y(x) = B^{\pm}_y + \Delta_x B^{\pm}_y(c(x) - c_0),$$

where $B_y^\pm$ are values for upper and lower coarse faces, $\Delta_x B_y^\pm$ are the limited slopes, c_0 is the face center for $[x_1, x_2]$, and $c(x)$ is the face center for $[x_1, x]$. For the face located at $x \in [x_1, x_2]$, the face component is defined by

$$B_x(x) \cdot A_x(x) = B_x^- \cdot A_x(x_1) + B_y^-(x) \cdot A_y^-(x) - B_y^+(x) \cdot A_y^+(x),$$

where $A_x(x)$ is the face area for the face $[y_1, y_2]$ located at x, $A_y(x)$ is the face area for the face $[x_1, x]$.

The cell-average value of B_x is

$$B_x = \frac{1}{V} \int_{x_1}^{x_2} (B_x(x) \cdot A_x(x)) dx.$$

For Cartesian grid, $c_0 = \frac{1}{2}(x_1 + x_2)$, $c_x = \frac{1}{2}(x_1 + x)$, $A_x(x) = y_2 - y_1 = \Delta y$, $A_y(x) = x - x_1$, and $V = \Delta x \cdot \Delta y$. The cell-centered value of B_x is

$$B_x = \frac{1}{V} \int_{x_1}^{x_2} \left(B_x^- \cdot \Delta y + \left((B_y^- - B_y^+) + (\Delta_x B_y^- - \Delta_x B_y^+) \frac{1}{2}(x - x_2) \right) (x - x_1) \right) dx.$$

After simplification and applying the divergence-free condition, we obtain

$$B_x = \frac{(B_x^- + B_x^+)}{2} + \frac{(\Delta x)^2}{12} \frac{\Delta_x B_y^+ - \Delta_x B_y^-}{\Delta y}. \tag{9}$$

Similarly, we can obtain the cell-centered value for B_y, which is

$$B_y = \frac{(B_y^- + B_y^+)}{2} + \frac{(\Delta y)^2}{12} \frac{\Delta_y B_x^+ - \Delta_y B_x^-}{\Delta x}. \tag{10}$$

Note that (9) and (10) are different from those obtained by Balsara (Balsara 2000). This reflects the difference between the cell-average value at the cell-center and field component values at the cell-center. For a second order method, the high order terms in (9) and (10) can be neglected, and the cell-centered values are the just the arithmetic average of the face-centered values.

4. AMR

We used the Berger-Colella (Berger & Colella 1989) AMR strategy in our code to handle those regions that need fine grid resolution. The detail of our implementation for general time-dependent PDEs was described in (Li & Hyman 1998a,b). To be concise, we here only describe the outline of our implementation.

Given a grid and solution on it, we first flag those cells that require refinement and then cluster them into several rectangular grids, called *patch.* For each newly-generated patch, we interpolate the solutions from the coarse grid to the new grid. If the old mesh already has some refinement that overlaps with the new mesh, the solution should be copied from the old fine mesh to the new one. The whole regridding procedure from the old mesh solution to the new mesh solution is called *prolongation.* If the initial conditions are defined analytically, the initial conditions on the new grid are also defined analytically. Each patch can be treated as a single grid. Further refinement can be done recursively until no refinement is needed or the finest refinement level is reached.

During the time integration, the coarse grid is integrated first, and then the fine grid. The boundary conditions of the fine grid are defined by either external boundary conditions, or adjacent sibling patches, or interpolations from the coarse parent grid. After the integration of the fine grid is done, we transfer the more accurate fine grid solutions to the parent coarse grid. This procedure is called *restriction.* The restriction could cause a loss of conservation at the interfaces between a fine and a coarse grid. A flux correction step is proposed in (Berger & Colella 1989) to maintain that the consistent fluxes are used at the fine-coarse interface.

Since the time stepsize is usually determined by the finest grid. Berger and Colella used a local time step method to improve the efficiency. This local time step method allows the fine grid takes smaller time step than the coarse grid before they reach the same time level.

We have enhanced the strategy of Berger and Colella with several features in (Li & Hyman 1998a,b). We allow the refinement to be controlled by a user-input grid file. We extended Berger's approach to method of lines (MOL) solver with high order time integrations. We improved the clustering algorithm by using a adaptive threshold. We also implemented the adaptation based on the geometry, the use of staggered grids and the choice of the refinement ratios.

4.1. Adaptation for geometry

In our implementation, the base grid can be either uniform grid, tensor-product grid, general curvilinear grid. For rectangular grid, we also allow the geometry to be Cartesian grid, Cylindrical grid, or Spherical polar grid. Although the basic algorithm is the same for the geometry, the implementation for prolongation and restriction is different for different geometries. We will describe the algorithm for cylindrical and spherical grid in section 6.

4.2. Staggered grid

The staggered grid is allowed in our implementation to handle the vector field. For finite-volume method, the solution can be put at cell-center, face-center, or edge center. For finite difference method, the solution may be node-centered. The staggered grid can pose difficulty in data structure and management in AMR framework, since each component of the solutions may have different length on a single grid. We propose to use a single pointer to represent the index location for a patch in AMR hierarchical data structure. For example, if a problem has n_u components on a $n_x \times n_y$ grid, the patch will be allocated a space equal to $n_x n_y n_u$ variables. In this way, some components of the solutions may not fill the whole patch. The cell-centered component only has $(n_x - 1)(n_y - 1)$ variables, whereas the face-centered component has $n_x(n_y - 1)$ variables in x-direction and has $(n_x - 1)n_y$ variables in y-direction. Only the node-centered component has $n_x n_y$ variables. This data structure wastes the storage a little bit. However it is easy to manage for staggered grid.

We also proposed a new approach to preserve the divergence-free vector field when the mesh is adapted. This new approach, which is described in (Li & Li 2003), works efficiently for any refinement ratio and any geometry.

4.3. Periodic boundary conditions for staggered grid

In a staggered grid, the field components may locate at the face. If an application has periodic boundary conditions, the field components located at the domain boundaries may be calculated twice in different patches in the AMR grids. It is important to treat the periodic boundaries just as the internal boundaries to ensure the conservation cross the domain boundaries. If a patch shares with a periodic boundary, or is within a distance of number of ghost zones to a periodic boundary, it can be potentially an updating partner for a patch on the other end. We solved this problem by adding another virtual patch, which is out of computational domain, on the other end, and copying the solution variables from the internal patch according the periodic boundary conditions.

4.4. Adaptation on time solver

One of important and useful features in our AMR framework is to allow different time solvers. As far as a solver for a single mesh is concern, the time integration can be widely different. It can be either dimensional splitting, which is common for Godunov solver, or dimensional unsplitting. For systems that contain source terms, an operator splitting method is also often used. The time integration can be method of lines (MOL) approach, where a semi-discretized ODE system is obtained first by spatial discretization and then higher order ODE solver is used. To accommodate the need for different solver, the traditional Berger's time integration approach must be changed.

4.4.1. Lax-Wendroff type time discretization

The Lax-Wendroff type time discretization uses high order spatial derivatives to replace time derivatives in order to achieve the high order accuracy. The second order accuracy method is usually used. This approach usually has a more compact stencil than the MOL multistage approach. Like the forward Euler method in time integration, this approach requires only the data at u_n to obtain the solution at u_{n+1}. Therefore it is easy to be implemented in an AMR framework.

4.4.2. Operator splitting method

The Lax-Wendroff type of scheme works efficient for 1-D scalar or system of equations. However, it is much more complicated for multi-dimensional problems. The dimensional splitting method is usually adopted for those cases. All of the splitting methods can pose difficulties for AMR framework. These methods are usually implemented via alternative direction methods (6) rather than the fractional step method (7), because the (7) is not only inefficient but also inaccurate, for the CFL number is only half of the requirement and small CFL number introduces more numerical diffusion and dispersion.

In Strang splitting (6), each procedure (two-steps) contains four sweeps for a 2-D problem, and the four sweeps should use the same time steps. This may pose a problem in treating each sweep as an independent small step in an AMR framework. The easiest way is to combine the four sweeps as one big step in an AMR framework. The difference between this one big step and any other one-step PDE solver for a single grid is that the communication between patches in the same level is required in each sweep in a big step.

Another difficulty with (6) is that the cosntraint transport on staggered grid we described in section 3.7 needs special treatment. Each sweep acts as a finite-volume method for cell-average variables. The boundary condition is defined no longer for the staggered grid but for cell-centered values. In order to compute the EMF at each edge (nodes in 2D case), the fluxes should be collected from each sweep in a larger region (one more extra zone than the interested region). The contraint transport can be taken as a postprocessing procedure after all of the other quantities have been calculated.

One potential drawback for this approach is that the internal boundary conditions for each fine patch may not be updated timely during the sweep if the data at the boundaries are obtained through interpolation from the coarse parent patch, because we did not save any intermediate information for the coarse sweep. Similarly, the coarse sweep may not updated timely with the most current data in the fine sweep. To overcome this problem, we propose a different approach. We used the locked time step method for this time solver. The locked time step means that all of the grids at different levels use the same time step and advance simultaneously to the next time level. One might argue that the very fine time steps used on the coarse grid introduces more numerical diffusion and dispersion. A counter argument to this, suggests that this can never be too serious because the reason that these grid blocks were coarsely refined was that there was not very much structure there anyway.

Because of the locked time step, we can update the boundary conditions for each fine patch and upate the data for each coarse patch with the most current information.

4.4.3. Multistage time discretization

This approach, namely the method of lines plus an ODE solver, has an advantage of simplicity, both in concept and in coding. Unlike the splitting approach, the multistage time discretization can achieve arbitrary order of accuracy, depending on the spatial discretization and order of the ODE solver. In this approach, it requires data communications between the AMR grids in each of the intermediate stages. The external boundary conditions also needs to be computed in each stage. Hence we cannot plug this type of solver directly into the AMR framework.

One interesting issue is how to interpolate the coarse parent grid solutions to get the boundaries data for the current level grid at intermediate stage. There are two approaches: one is to do both time and space interpolation with the coarse parent data saved at t_n and t_{n+1}, the other is to save a copy of the intermediate state of the coarse parent data and do only the space interpolation. We had compared the two approaches and found they produce almost the same results. Since the second approaches requires more storage, we adopt the first one in our AMR framework. We also improve the first approach that only a space interpolation is required for only the first stage. The interpolation is applied to the time derivatives du/dt rather than to the solution u. Therefore, when the interpolation is needed for u at intermediate stage, it can be easily calculated.

Another problem with the multistage time discretization is how to calculate the flux at the interface, which is required for flux conservation by a correction step on the coarse parent flux. We cannot take the flux at the final stage as the flux for the whole time step. It is also inefficient to do the flux correction for each intermediate stage. We instead obtain the flux for the whole time step by composing the flux at each stage together. The composing is not simply a sum over all of the stages. Take the third order TVD Runge-Kutta solver (Shu & Osher 1989) as an example. The time discretization contains three stages,

$$\begin{aligned}
u^{(1)} &= u^n + \Delta t L(u^n, t^n), \\
u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L(u^{(1)}, t^n + \Delta t), \\
u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L(u^{(2)}, t^n + \frac{1}{2}\Delta t),
\end{aligned}$$

where $L(u,t)$ is a spatial discretization operator. As we can see, each stage has a contribution to the total flux of the whole time step. The total flux should be

$$f_{n,n+1} = \frac{1}{6}f(u^n, t^n) + \frac{1}{6}f(u^{(1)}, t^n + \Delta t) + \frac{2}{3}f(u^{(2)}, t^n + \frac{1}{2}\Delta t),$$

where $f(u,t)$ the flux evaluation at each stage.

5. Parallel

Our AMR implementation is also parallelized via MPI approach so that it can be used for parallel computation with a large number of processors. To improve the efficiency of our parallelization, we modify our AMR implementation in several aspects.

5.1. Locked time step versus local time step

We have compared the locked time step versus the local time step method in (Li & Hyman 2002), and found that the local time step method is only slightly (about 50%) better than the locked time step for a 2-D problem with refinement area more than 8%. The local time step method can potentially become a serial bottle-neck for parallelization with a larger number of the processors, because each level must be done sequentially with the coarse level first. However, for the locked time step method, all of the grids at different levels can advance simultaneously. The local time step also requires extra memory to be allocated to store the temporary copies of the solutions.

We should pointed out that not every time integration can be done simultaneously for all the grids at different levels. For high order MOL approach, cf., 3rd order Runge-Kutta time integration, the boundary conditions are required for intermediate step during a full-step integration, which might require interpolations in time and space from the coarse parent grid. Then the grids on each level must be advanced sequentially.

5.2. Clustering

We have adopted Berger and Rigoutsos (Berger & Rigoutsos 1991) clustering strategy with a adaptive threshold. It is fast and efficient for serial computation. However, it is done at an entire level so that it is difficult to parallelize. Apart from that, clustering at an entire level also requires too much communications for parallel machines with distributed-memory. For these reasons, we have applied the clustering to each of the single-grid at a level ,which is easily parallelized. It might produce more single grids than one might obtained if one had used the clustering at an entire level. However, it does ensure that all the flagged cells are fully covered. Moreover, the grid generated by this clustering approach has only one parent, which may simplify the data structure and reduce the communications between different levels.

5.3. Load balancing

We have two types of load balancing in our implementation: global load balancing and local load balancing for each level. If the time integration can be synchronized and the number of the processors may be larger than the number of the patches in any fine level, a global load balancing should be used to utilize the power of all of the processors. On the other hand, if the number of processors is less than the number of the patches in any level, the local balancing for each level may be used to improve the efficiency.

Given a number of grids, we have two concerns when doing the load balancing. The first is to ensure that each processor has the same amount of load so that they can finish the computational tasks at roughly the same time. The other concern is to reduce the communication between different levels to as small as possible. There is a trade-off between these two concerns.

One possibility arise that all of the grids have the same size to make it easy for load-balancing. This strategy has been used by PARAMESH (Macneice et al. 2000). PARAMESH has been a key component in several MHD and HD codes including FLASH code at University of Chicago and BATSRUS at University of Michigan. However, when the number of the grids is not divisible by a total number of the processors, which is very common in scientific computation, the load will be out of load balance. For that reason, we do not require all the grids to have the same size. Moreover, our load balancing is designed so that it actually capitalizes on this difference sizes to achieve precise load balance.

Ideally, we wish to achieve that the load is balanced among the processors and the inter-processor communication between different levels is minimized to zero. However, to achieve this goal, it might require more communications in load balancing procedure. DAGH (Mitra et al. 1997) used a composite decomposition of the adaptive grid hierarchy to achieve zero inter-grid communications. The composite decomposition requires re-distribution that requires more data movement. In our implementation, we adopted a simple approach. We first assigned that each fine grid share the same processor with its parent. Since each grid has only one parent, this step can be easily done. Then we balance the load among different processors by moving and splitting the patches: We calculate the load for each processor and move the patches from the processor with the maximum load to the one with the minimum load under the condition that the computational load of the moving patch is no more than the difference of the average load and minimum load. This phase continues until either the load is balanced or patches are too large to be moved. If imbalance still exists, we then find the largest patch in the processors with the maximum load and split it into two patches along the longest dimension. One of the two split patches should have size that could make the processor with minimum load reach the balanced load. These two steps execute in parallel until the load is balanced among all of the processors.

We note that by moving and splitting patches the future inter-grid communication might increase because the moving grid has different processor with its parent. InAn effort is being done to reduce the communication in the moving phase.

6. Spherical and Cylindrical Geometry

6.1. Discretization

By introducing general curvilinear coordinates, we have also introduced the possibility of numerical errors when differencing near coordinate singularities. The occurrence of this error can be easily be demonstrated in spherical polar coordinates. For radial derivatives, we need to difference

$$\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 F). \tag{11}$$

There are three central differencing,

$$\begin{aligned}
\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 F) &\approx \frac{\Delta(r^2F)}{r^2\Delta r} = \frac{r_i^2F_i - r_{i-1}^2F_{i-1}}{[(r_i+r_{i-1})/2]^2(r_i-r_{i-1})}, && (12)\\
\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 F) &\approx \frac{\Delta(r^2F)}{\Delta(r^3/3)} = \frac{r_i^2F_i - r_{i-1}^2F_{i-1}}{r_i^3/3 - r_{i-1}^3/3}, && (13)\\
\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 F) &\approx \frac{2F}{r} + \frac{\Delta F}{\Delta r} = \frac{4F_{i-\frac{1}{2}}}{r_i+r_{i-1}} + \frac{F_i - F_{i-1}}{r_i - r_{i-1}}, && (14)
\end{aligned}$$

Although all of these difference approximations are almost identical at large radii, near the origin ($r_{i-1} \to 0$) there is a serious discrepancy between them. To see the difference, we assume that $r_{i-1} = 0$. Then equations (12) to (14) become

$$\begin{aligned}
\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 F) &\approx \frac{4F_i}{r_i}, && (15)\\
\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 F) &\approx \frac{3F_i}{r_i}, && (16)\\
\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 F) &\approx \frac{4F_{i-\frac{1}{2}} + F_i - F_{i-1}}{r_i}. && (17)
\end{aligned}$$

If $F = c$ is constant, then

$$\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 F)|_{r=r_{i-\frac{1}{2}}} = \frac{2c}{r_{i-\frac{1}{2}}} = \frac{4c}{r_i},$$

which means that (12) and (14) is accurate while (13) has large error by c/r_i. If $F = r$ is a linear function, then

$$\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 F) = 3,$$

which means (13) and (14) is accurate while (12) has large error by 1. If $F = r^2$ is a quadratic function, then

$$\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 F)|_{r=r_{i-\frac{1}{2}}} = 4r_{i-\frac{1}{2}} = 2r_i,$$

which means only (14) is correct, and (12) has error by $2r_i$, (13) has error by r_i. If $F_{i-\frac{1}{2}}$ is approximated by

$$F_{i-\frac{1}{2}} = \frac{1}{2}(F_i + F_{i-1}),$$

then even (14) has error by r_i. Because r_i is close to the origin, we can assume that $r_i \ll 1$. Then the error c/r_i can be large. Therefore (13) is not recommended for use near the origin in a finite-difference method.

If a finite-volume (or control volume) method is used, we need to integrate (11) and then divide it by the volume difference, which yields

$$\frac{\int dr \int d\theta \left[\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 F) r^2 \sin\theta\right]}{\int dr \int d\theta (r^2 \sin\theta)} \approx \frac{r_i^2 F_i - r_{i-1} F_{i-1}}{r_i^3/3 - r_{i-1}^3/3},$$

which is exactly the finite difference approximation (13). However, it has different meaning. In the finite-volume sense, (13) means the approximation to the volume average value of (11), whereas in the finite-difference sense, it means the point value at the center of the volume. After carefully calculation, we found that for $F = c$, $F = r$ and $F = r^2$, (13) has no error. Therefore, if the finite-volume is to be used, only (13) is recommended.

Similar problems occur from differencing the polar angle derivatives, where we approximate

$$\frac{1}{r\sin\theta}\frac{\partial \sin\theta G}{\partial\theta}.$$

Using the same finite-volume method, we obtain

$$\frac{\int dr \int d\theta \left[\frac{1}{r\sin\theta}\frac{\partial \sin\theta G}{\partial\theta} r^2 \sin\theta\right]}{\int dr \int d\theta (r^2 \sin\theta)} = \frac{(G_j \sin\theta_j - G_{j-1}\sin\theta_{j-1})}{\cos\theta_{j-1} - \cos\theta_j}\frac{\frac{1}{2}r_i^2 - \frac{1}{2}r_{i-1}^2}{\frac{1}{3}r_i^3 - \frac{1}{3}r_{i-1}^3}$$

The source term should also be approximated by the same method as other terms. For spherical polar coordinates, the source term is

$$S = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \end{pmatrix} = \left(\begin{array}{c} 0 \\ (\rho(v_\theta^2 + v_\phi^2) + 2p + B_r^2)/r \\ (\rho(v_\phi^2 cot\theta - v_\theta v_r) + B_r B_\theta + (p + \frac{1}{2}(B_r^2 + B_\theta^2 - B_\phi^2))cot\theta)/r \\ (-\rho(v_\phi v_\theta cot\theta + v_\phi v_r) + B_\phi B_r + B_\phi B_\theta cot\theta)/r \\ 0 \\ \Omega_3/r \\ (\Omega_1 cot\theta - \Omega_2)/r \\ 0 \end{array} \right) \tag{18}$$

If the finite volume method is used, the source term should be approximated in a finite-volume sense. That is to replace $cot\theta$ with

$$\frac{\int dr \int d\theta (cot\theta r^2 \sin\theta)}{\int dr \int d\theta (r^2 \sin\theta)} = \frac{\sin\theta_j - \sin\theta_{j-1}}{\cos\theta_{j-1} - \cos\theta_j},$$

and replace $1/r$ with

$$\frac{\int dr \int d\theta (\frac{1}{r} r^2 \sin\theta)}{\int dr \int d\theta (r^2 \sin\theta)} = \frac{\frac{1}{2}r_i^2 - \frac{1}{2}r_{i-1}^2}{\frac{1}{3}r_i^3 - \frac{1}{3}r_{i-1}^3},$$

in (18). Approximating (18) with the cell-centered values of r and θ directly will result in large error near origin.

6.2. Staggered grid and magnetic field components

To maintain the divergence free conditions of magnetic field components, we treat them as the face-average variables located at the face-center, and hence a staggered grid has to be used. The divergence of a vector field and its discretization for the cylindrical and spherical coordinates is different from that for the Cartesian grid. In 2-D (r, θ) of the spherical coordinates, the divergence condition becomes

$$\nabla \cdot B = \frac{1}{r^2}\frac{\partial(r^2 B_r)}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial(\sin\theta B_\theta)}{\partial\theta}. \tag{19}$$

The evolution equations for B_r and B_θ are

$$\frac{\partial B_r}{\partial t} = \frac{1}{r\sin\theta}\frac{\partial(\sin\theta\Omega_3)}{\partial\theta}, \tag{20}$$

$$\frac{\partial B_\theta}{\partial t} = -\frac{1}{r}\frac{\partial(r\Omega_3)}{\partial r}. \tag{21}$$

We again have three options to discretize (19), (20) and (21), as indicated in (12) to (14). For the staggered grid, B_r and B_θ are face-centered variables that have face average values. Integrating (19) for the cell (i, j) we obtain,

$$\nabla \cdot B = \frac{1}{\Delta V}((B_{r_i} r_i^2 - B_{r_{i-1}} r_{i-1}^2)(\cos\theta_{j-1} - \cos\theta_j) + B_{\theta_j} r_{i-\frac{1}{2}} \sin\theta_j dr - B_{\theta_{j-1}} r_{i-\frac{1}{2}} \sin\theta_{j-1} dr) \tag{22}$$

where $\Delta V = (\frac{1}{3}r_i^3 - \frac{1}{3}r_{i-1}^3)(\cos\theta_{j-1} - \cos\theta_j)$ (where we drop $d\phi$ in our formulations) is the cell volume. Note that $r_i^2(\cos(\theta_{j-1} - \cos\theta_j)$ is the area of the face $(i, j - \frac{1}{2})$, and $r_{i-\frac{1}{2}} \sin\theta_j dr$ is the area of the face $(i - \frac{1}{2}, j)$, the discretization is in good agreement with the physical definition of the divergence. At the origin $r = 0$, we have only three faces, and hence we do not need to worry about the value of B_r at the origin.

The equations of (20) and (21) should be discretized in such a way that if (22) is zero at $t = t_n$, it remains zero at $t = t_{n+1}$. Following Yee's method for the Cartesian grid, we discretize (20) and (21) as follows

$$\frac{B_{r_i}^{n+1} - B_{r_i}^n}{\Delta t} = \frac{\sin\theta_j \Omega_{i,j} - \sin\theta_{j-1}\Omega_{i,j-1}}{r_i(\cos\theta_{j-1} - \cos\theta_j)}, \tag{23}$$

$$\frac{B_{\theta_j}^{n+1} - B_{\theta_j}^n}{\Delta t} = -\frac{r_i \Omega_{i,j} - r_{i-1}\Omega_{i-1,j}}{r_{i-\frac{1}{2}} dr}. \tag{24}$$

Note that we use $(\cos\theta_{j-1} - \cos\theta_j)$ rather than $\sin\theta_{j-\frac{1}{2}} d\theta$ in the discretization of B_r to maintain that (22) is zero.

ZEUS code (Stone & Norman 1992) used $\sin\theta_{j-\frac{1}{2}} d\theta$ to discretize $\sin\theta d\theta$ in (22) and (20). We think our discretization is more accurate than theirs as regard the area of the interface.

6.3. Singularity of magnetic field at origin

As we have mentioned, at origin, B_r has no unique solutions. The initial B_r at the origin can be calculated by a potential method. We are concerning how to advance B_r at the origin. Although it is not used in the calculation of (22), it is needed in calculating the cell-centered value of B_r. Therefore, we cannot assign a nominal value to it at a given time.

Suppose that all of B_r and B_θ are known except the B_r at origin. We can use (14) to discretize (19) and obtain

$$\frac{4B_{r_{\frac{1}{2}}} + B_{r_1} - B_{r_0}}{r_1} + \frac{\sin\theta_j B_{\theta_j} - \sin\theta_{j-1} B_{\theta_{j-1}}}{r_{\frac{1}{2}}(\cos\theta_{j-1} - \cos\theta_j)} = 0.$$

Let $B_{r_{\frac{1}{2}}} = \frac{1}{2}(B_{r_0} + B_{r_1})$, we obtain

$$B_{r_0} = -3B_{r_1} - 2\frac{\sin\theta_j B_{\theta_j} - \sin\theta_{j-1} B_{\theta_{j-1}}}{(\cos\theta_{j-1} - \cos\theta_j)}.$$

This approach cannot be used in a cylindrical coordinate. An alternative approach is to use the B_r calculated at the cell-centered by the Godunov method and the value of B_{r_1} to interpolate the value B_{r_0}.

6.4. Boundary conditions

For the spherical coordinates, the range of θ is $[0, \pi]$, the boundary at any $r = r_0$ is sphere surface. At origin, $r = 0$, the sphere is degenerated into a point. Although, $r < 0$ makes no sense in reality, the ghost cell at the origin does have meaning. The values of the ghost cell at $(-r, \theta, \phi)$ are defined by the values at $(r, \pi - \theta, \phi + \pi)$. For the vector field $\mathbf{v}$ and $\mathbf{B}$, $\vec{e}_\theta$ remains the same, whereas $\vec{e}_r$ and $\vec{e}_\phi$ have opposite directions. Hence, the density, energy, velocity and magnetic component in θ direction at $(-r, \theta, \phi)$ should take the same values as those at $(r, \pi - \theta, \phi + \pi)$, while the velocity and magnetic components in r and ϕ directions should take the negative of values in the corresponding active zones. This boundary condition is identity to the reflecting boundary conditions for Cartesian grid with the reflecting interface to be θ except that the vector field component in ϕ direction is set to negative of the value in the corresponding active zones.

Similarly the boundary conditions at $\theta = 0$ and $\theta = \pi$ can be derived in the same way. At any $\theta = \theta_0$, the boundary is a cone surface. However, the cone is reduced to a line at $\theta = 0$ and $\theta = \pi$. At $\theta = 0$ boundary, the values of ghost cell at $(r, -\theta, \phi)$ are defined by the values at $(r, \theta, \phi + \pi)$. At $\theta = \pi$ boundary, the values of ghost cell at $(r, \theta + \pi, \phi)$ are defined by the values at $(r, \pi - \theta, \phi + \pi)$.

6.5. Prolongation and restriction for AMR

Since the geometry has changed, the prolongation and restriction for Cartesian AMR grids may not work for the cylindrical or spherical grid. In this subsection, we describe the prolongation and restriction operations for cell-centered variables. The operations for face-centered variables will be described in the next subsection.

We assume that all of the variables resided at the cell center and are the cell-average values over the whole cell. We should pointed out that our approach is different from that in (Pember et al. 1996).

6.5.1. Restriction

We first describe the restriction operation. The restriction should make a volume weighted average of a variable and assign it to the corresponding cell of the coarse mesh. For Cartesian grid, it can be done by simple arithmetic averaging over all of the fine cells because each fine cell has the same volume. For cylindrical and spherical grid, it may not be true that each fine cell has the same volume. The volume of a cylindrical cell is

$$\int_{z_1}^{z_2} dz \int_{r_1}^{r_2} dr \int_{\theta_1}^{\theta_2} r d\theta = \frac{1}{2}(r_2^2 - r_1^2)\Delta\theta\Delta z.$$

The volume of a spherical cell is

$$\int_{r_1}^{r_2} dr \int_{\theta_1}^{\theta_2} r d\theta \int_{\phi_1}^{\phi_2} r \sin\theta d\phi = \frac{1}{3}(r_2^3 - r_1^3)(\cos\theta_1 - \cos\theta_2)\Delta\phi.$$

It is easy to see that the volume depends on r for a cylindrical grid, and depends on r and θ for a spherical grid. The volume-weighted average value is no longer a simple arithmetic average.

Consider a reduced 2-D (r, θ) problem (see Fig. 6.1). We can drop Δz or $\Delta\phi$ in the volume formula. If the refinement ratio is 2 and the values on the fine grid is $u_{f_{i,j}}$, the volume-weighted average value for the cylindrical grid is then

$$u_c = \frac{(u_{f_{0,0}} + u_{f_{0,1}})dA_0 + (u_{f_{1,0}} + u_{f_{1,1}})dA_1}{dA_c},$$

where $dA_0 = (r_1^2 - r_0^2)$, $dA_1 = (r_2^2 - r_1^2)$, and $dA_c = (r_2^2 - r_0^2)$.

The volume average value for the spherical grid is

$$u_c = \frac{u_{f_{0,0}} dA_{0,0} + u_{f_{0,1}} dA_{0,1} + u_{f_{1,0}} dA_{1,0} + u_{f_{1,1}} dA_{1,1}}{dA_c},$$

where

$$\begin{aligned} dA_{0,0} &= (r_1^3 - r_0^3)(\cos\theta_0 - \cos\theta_1), \\ dA_{0,1} &= (r_1^3 - r_0^3)(\cos\theta_1 - \cos\theta_2), \\ dA_{1,0} &= (r_2^3 - r_1^3)(\cos\theta_0 - \cos\theta_1), \\ dA_{1,1} &= (r_2^3 - r_1^3)(\cos\theta_1 - \cos\theta_2), \\ dA_c &= (r_2^3 - r_0^3)(\cos\theta_0 - \cos\theta_2), \end{aligned}$$

6.5.2. Prolongation

The prolongation is far more intricate than the restriction problem. The difficulty is to find a conservative interpolation. We here study only the monotone-preserved linear

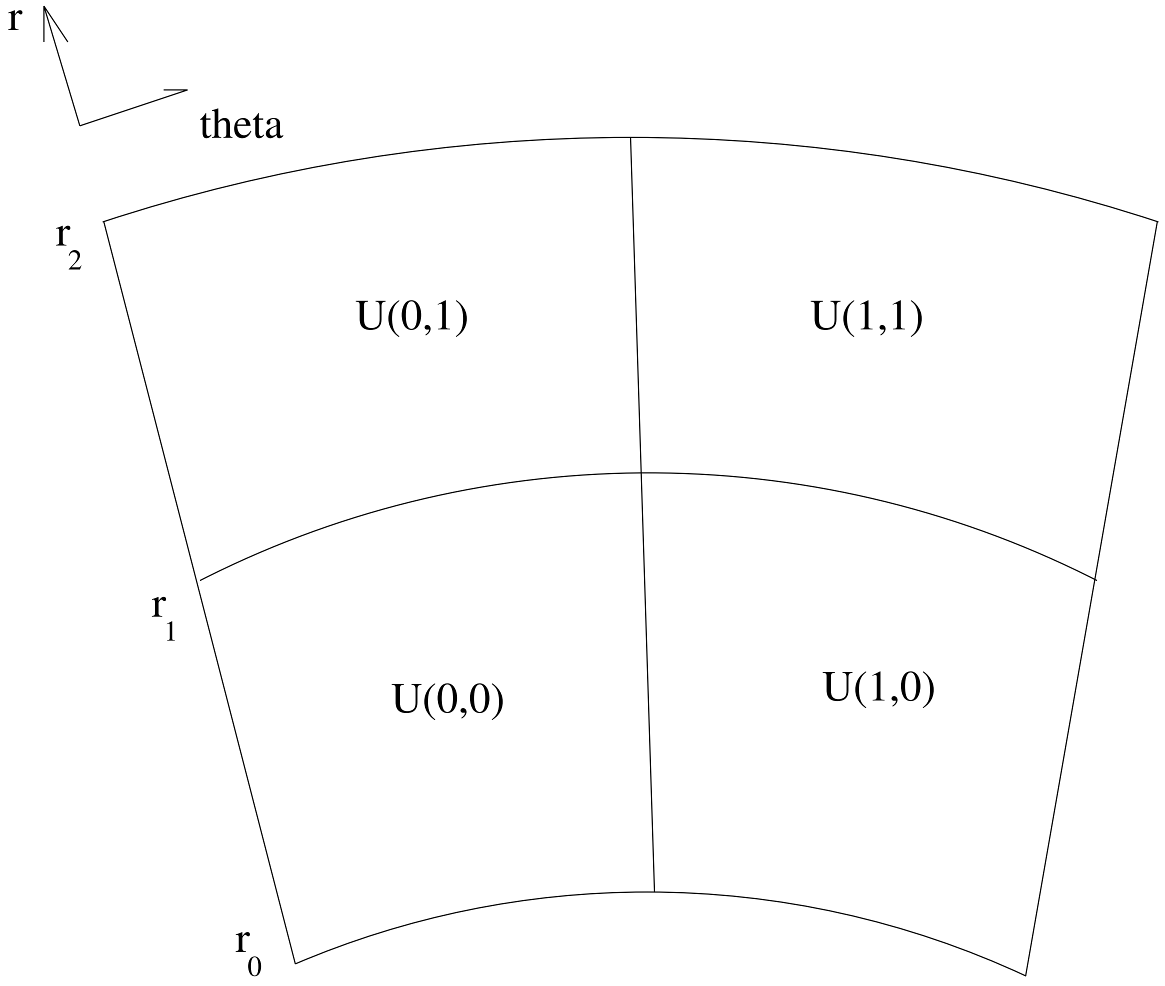

Fig. 6.1.— A reduced 2D (r, θ) coordinates

interpolation. For a reduced 2-D (r, θ) problem. The interpolation on a coarse cell $[r1, r2] \times [\theta_1, \theta_2]$ can be formulated as

$$u(r, \theta) = u_c + \Delta_r u_c(r - r_c) + \Delta_\theta u_c(\theta - \theta_c), \tag{25}$$

where Δ_r and Δ_θ are limited slopes in corresponding r and θ direction, (r_c, θ_c) is the location of the cell center.

It is naive to think that the cell center locates at $\frac{1}{2}(r_1 + r_2), \frac{1}{2}(\theta_1 + \theta_2)$ for cylindrical and spherical grid. By definition, the cell center should satisfy the following condition

$$\int_{r_1}^{r_2} dr \int_{\theta_1}^{\theta_2} u(r, \theta) r dr d\theta = u_c \cdot dA_c, \tag{26}$$

for the cylindrical grid, and

$$\int_{r_1}^{r_2} \int_{\theta_1}^{\theta_2} u(r, \theta) r^2 \sin\theta dr d\theta = u_c \cdot dA_c, \tag{27}$$

for the spherical grid.

For a cylindrical grid, we can obtain the location of the cell center by inserting Eq. (25) into (26), which yields after simple manipulation,

$$\begin{aligned} r_c &= \frac{2}{3} \frac{(r_2^3 - r_1^3)}{(r_2^2 - r_1^2)}, && (28) \\ \theta_c &= \frac{1}{2}(\theta_2 + \theta_1). && (29) \end{aligned}$$

For a spherical grid, the location of the cell center can be obtained by inserting Eq. (25) into (27),

$$\begin{aligned} r_c &= \frac{3}{4} \frac{(r_2^4 - r_1^4)}{(r_2^3 - r_1^3)} && (30) \\ \theta_c &= \frac{\theta_1 \cos\theta_1 - \theta_2 \cos\theta_2 + \sin\theta_2 - \sin\theta_1}{\cos\theta_1 - \cos\theta_2}. && (31) \end{aligned}$$

The locations of the cell-center for the fine grid can be calculated in the same way. Then the cell-average values for the fine grid are calculated by (25). We have verified that the cell-centered values obtained by (25) with r_c and θ_c calculated by (28)–(31) for find grid cells satisfy the conservation laws.

6.6. Maintaining divergence free conditions for AMR

The maintenance divergence free conditions for AMR requires some modifications when it is applied to the polar spherical grid.

We first study the restriction operation in AMR framework. It consists of making an area weighted average of the magnetic field component that is collocated on the faces of the fine mesh and assigning it to the corresponding face of the coarse mesh. Such step should be carried out, along with the electric field correction step that is discussed later in this section, whenever the fine and coarse meshes are temporally synchronized. It can be shown easily that if the field on the fine mesh is divergence free, the combination of area-weighted restriction and electric field correction will result in a divergence free field on the coarse mesh.

For the 2-D (r, θ) polar spherical coordinates, it is actually a 3-D problem with symmetry in ϕ direction, and hence the area of the face is quite different from that of Cartesian or cylindrical coordinates. As we have mentioned in Section 6.2, the area for a cell $(i, j-\frac{1}{2})$ located at (r, ϕ) face is $r_i^2(\cos\theta_j - \cos\theta_{j-1})$ (as usual, we drop $d\phi$ term), and the area for a cell $(i-\frac{1}{2}, j)$ located at (θ, ϕ) face is $\sin\theta_j(\frac{1}{2}r_i - \frac{1}{2}r_{i-1})$. The area-weighted restriction applied to the magnetic field component then yield (assume the refinement ratio is 2)

$$B_{r_c} = \frac{B_{r_1} dA_{r_1} + B_{r_2} dA_{r_2}}{dA_{r_1} + dA_{r_2}} \tag{32}$$

$$B_{\theta_c} = \frac{B_{\theta_1} dA_{\theta_1} + B_{\theta_2} dA_{\theta_2}}{dA_{\theta_1} + dA_{\theta_2}} \tag{33}$$

where B_{r_i} and B_{θ_i} are the values on the fine grid, and dA_{r_i} and dA_{θ_i} are the area at the interfaces, which is defined by

$$\begin{aligned} dA_{r_i} &= r_i^2(\cos\theta_j - \cos\theta_{j-1}), \\ dA_{\theta_j} &= \sin\theta_j(\frac{1}{2}r_i - \frac{1}{2}r_{i-1}). \end{aligned}$$

Although the mesh is uniform on (r, θ) plane, dA_{r_i} (and dA_{θ_i}) is different for different i.

The electric field correction should be done at any fine-coarse interface. This ensure that a coarse mesh that shares a face with the fine mesh will always have the best possible estimate for the magnetic field when the fine and coarse meshes are temporally synchronized. For a reduced 2-D (r, θ) polar spherical coordinates, the electric field has only one component (Ω_3) that locates at the node center. The correction should be made according to the equations (23) and (24) with the newly updated electric field. In fact, what is required during the correction step is only the difference of the electric field between the coarse value and fine value.

The prolongation operation from the coarse mesh to the fine mesh is far more intricate. First a divergence-free preserving interpolation must be used to obtain the values on the fine grid. A divergence free reconstruction scheme for Cartesian grid has been given by Balsara. We here extend it to the polar spherical grid. We found that Balsara's method cannot be applied to the spherical geometry directly, because the divergence equation (19) contains nonlinear functions $\sin\theta$. If we use a quadratic representation for B_r and B_θ, as was done by Balsara, we cannot get anything useful to maintain the divergence free conditions (22).

Another attempt is to rewrite (19) as

$$\nabla \cdot B = \frac{1}{r^2 \sin\theta}\left(\frac{\partial(r^2 B_r \sin\theta)}{\partial r} + \frac{\partial(r \sin\theta B_\theta)}{\partial\theta}\right).$$

If we denote $\tilde{B}_r = r^2 B_r \sin\theta$ and $\tilde{B}_\theta = r\sin\theta B_\theta$, then from $\nabla \cdot B = 0$, we have

$$\frac{\partial \tilde{B}_r}{\partial r} + \frac{\partial \tilde{B}_\theta}{\partial \theta} = 0, \tag{34}$$

which is similar to the divergence free condition in Cartesian grid. The Balsara's method for Cartesian grid can be used to reconstruct new $\tilde{B}_r$ and $\tilde{B}_\theta$ on the fine grid. This approach has several drawbacks. First, the discretization of (34) in Cartesian grid will not be the same as (22). There is a slight difference in calculating the area of the face at $(i, j - \frac{1}{2})$. The finite difference of (34) in Cartesian grid corresponds to the discretization of (19) with area of of the face at $(i, j - \frac{1}{2})$ to be $\sin\theta_{j-\frac{1}{2}}(\theta_j - \theta_{j-1})$. As we have mentioned, (22) is more accurate in finite-volume sense. Second, because the scaling factors become zeros when when $\sin\theta = 0$, we may not get the reconstructed B_r and B_θ in those cases. Thirdly the Balsara's approach (linear profile at the face) can be used only for the refinement ratio of two. This is because linear interpolation of the coarse face value to be matched by the fine values works only when the refinement ratio is two. Although it can be extended to arbitrary refinement ratio, it is much more complicated due to the high order interpolation polynomial. This drawback exists also for Cartesian grid. In our implementation, we used a new approach which was proposed in (Li & Li 2003) to reconstruct the divergence free field for the refined grid. The new approach is done in dimensional splitting manner and can be applied to any kind of grid with any refinement ratio.

To be used by nonuniform grid, we reconstruct the flux instead of the field itself. In case of zero area of flux face, we reconstruct the field itself at the specific face. We assume that all of the zero-area faces reside on the coarse grid, which is trivial for spherical grid because only the first $\theta = 0$ face and the last $\theta = \pi$ face have zero area.

To be concise, we describe our algorithm in θ direction for the reduced 2-D spherical grid. Suppose that the refinement ratio is n in θ direction and the coarse zone has faces at θ_0, θ_n, r_0 and r_1, we attempt to reconstruct the face values at θ_i $(0 < r < n)$. Due to the divergence free conditions, we only need reconstruct the field at the zone faces r_0 and r_1. This can be done either by the linear interpolations of the coarse value or by copying the fine field value directly if the face shares with a old fine mesh. Unlike the Balsara's approach, we can use any order of interpolation polynomial. For simplicity and the main method is of second order, we use linear interpolation. For a TVD reconstruction, any limiters can be used for the slope. To avoid flatting by the scaling factors, the limiters should be applied directly to B_r. After all of B_r at the zone faces have been calculated, the B_θ can be calculated sequentially by the divergence free condition.

For r direction, the same method can be used except that the new generated interfaces in θ direction need to be refined too. For the TVD reconstruction, the limiter can either be applied directly to the new interfaces or have the average value of those at the coarse interfaces.

The reconstruction at the zone faces can be done easily with monotonic-preserving reconstruction of B_r or B_θ. Let $B_r^+ = B_{r_{i,j}}$ denotes the magnetic field at the cell's upper r face and $B_r^- = B_{r_{i-1,j}}$ denotes the magnetic field at the cell's lower r face. Likewise, let $B_\theta^+ = B_{\theta_{i,j}}$ denotes the magnetic field at the cell's upper θ face and $B_\theta^- = B_{\theta_{i,j-1}}$ denotes the magnetic field at the cell's lower θ face. We can construct a piecewise linear profile via a slope limiter for $B_r^\pm$ and $B_\theta^\pm$. Assume the limited slope for $B_r^\pm$ is $\Delta_\theta B_r^\pm$, and the linear profile is

$$B_r^\pm = B_{r_c}^\pm + \Delta_\theta B_r^\pm (\theta - \theta_0).$$

Then the following condition should be satisfied

$$\frac{1}{r(\cos\theta_{j-1} - \cos\theta_j)} \int_{\theta_{j-1}}^{\theta_j} B_r^\pm \cdot r \sin\theta d\theta = B_{r_c}^\pm,$$

which yields

$$\theta_0 = \frac{\theta_{j-1}\cos\theta_{j-1} - \theta_j \cos\theta_j + \sin\theta_j - \sin\theta_{j-1}}{\cos\theta_{j-1} - \cos\theta_j}.$$

Similarly, for $\Delta_\theta B_r^\pm$ and linear profile

$$B_\theta^\pm = B_{\theta_c}^\pm + \Delta_r B_\theta^\pm (r - r_0),$$

and condition

$$\int_{r_{i-1}}^{r_i} (r - r_0) r \sin\theta dr = 0,$$

we can obtain

$$r_0 = \frac{2(r_i^3 - r_{i-1}^3)}{3(r_i^2 - r_{i-1}^2)}.$$

Note that θ_0 and r_0 are the face center for the corresponding face for the coarse grid. When the mesh is refined with ratio m, face center for the fine mesh can be calculated in the same fashion. Let $\delta r = (r_i - r_{i-1})/m$ and $r = r_i$, then the face center for the fine grid will be

$$r_{0k} = \frac{2((r - k\delta r)^3 - (r - (k+1)\delta r)^3)}{3((r - k\delta r)^2 - (r - (k+1)\delta r)^2)}, \quad k = 0, 1, ..., m - 1.$$

After the face center is calculated, the magnetic field components can be calculated by the linear profile

$$B^{\pm}_{\theta_{0k}} = B^{\pm}_{\theta_c} + \Delta_r B^{\pm}_{\theta}(r_{0k} - r_0).$$

It can be verified that

$$B^{\pm}_{\theta_{0k}} \cdot dA_{0k} = B^{\pm}_{\theta} \cdot dA,$$

which means the flux is conserved after interpolation. Similarly, we can obtain the face center θ_{0k} and obtain the B_r for the fine face.

In case of $dA_r = 0$ (e.g., $\theta = 0$ or $\theta = \pi$), we let $dA_r = \Delta r$, which is reduced to the reconstruction on a uniform 1-D grid. We should mentioned that if the new fine grid shares face with the old fine grid, the values of the magnetic field on the old fine grid should be copied to the new grid.

6.7. From face-centered value to cell-centered value

The overall algorithm has been described in section 3.8. For the cylindrical and spherical polar grid, we replace (x, y) with (r, θ) in equations in section 3.8.

6.7.1. Cylindrical grid

For Cylindrical grid, $c_0 = \frac{1}{2}(r_1 + r_2)$, $c_r = \frac{1}{2}(r_1 + r)$, $A_r(r) = r\Delta\theta$, $A_\theta(r) = (r - r_1)$, and $V = \frac{1}{2}(r_2^2 - r_1^2)\Delta\theta$. The cell centered value of B_r is

$$B_r = \frac{(B_r^- + B_r^+)}{2} + \frac{(\Delta r)^2}{12} \frac{\Delta_r B_\theta^+ - \Delta_r B_\theta^-}{\frac{1}{2}(r_1 + r_2)\Delta\theta}.$$

The cell centered value of B_θ is

$$B_\theta = \frac{(B_\theta^- + B_\theta^+)}{2} + \frac{(\frac{1}{2}(r_1 + r_2)\Delta\theta)^2}{12} \frac{\Delta_\theta B_r^+ - \Delta_\theta B_r^-}{\Delta r}.$$

Note that B_r and B_θ are similar to those for Cartesian grid with $\Delta y = r\Delta\theta$.

6.7.2. Spherical grid

The case for spherical grid is much more complicated than the previous two. It reflects a fully 3-D reconstruction. As we have derived,

$$c_0 = \frac{2}{3}\frac{r_2^3 - r_1^3}{r_2^2 - r_1^2},$$

$$c_r = \frac{2}{3}\frac{r^3 - r_1^3}{r^2 - r_1^2},$$

$$A_r(r) = r^2(\cos\theta_2 - \cos\theta_1),$$

$$A_\theta(r) = \frac{1}{2}(r^2 - r_1^2)\sin\theta,$$

and

$$V = \frac{1}{3}(r_2^3 - r_1^3)(\cos\theta_1 - \cos\theta_2).$$

From these equations, we can derive the cell-centered value of B_r to be

$$\begin{aligned} B_r \quad &= \quad \frac{1}{r_1^2 + r_2^2 + r_1 r_2}\left(B_{r_1} r_1^2 + B_{r_2} r_2^2 + \frac{r_1 r_2}{r_1 + r_2}(B_{r_2} r_2 + B_{r_1} r_1)\right) \\ &+ \quad \frac{(\Delta r)^2}{12}\frac{\Delta_r B_\theta^+ \sin\theta_2 - \Delta_r B_\theta^- \sin\theta_1}{\cos\theta_1 - \cos\theta_2}\frac{r_2^2 + r_2^2 + 4r_1 r_2}{r_2^3 + 2r_2^2 r_1 + 2r_2 r_1^2 + r_1^3} \end{aligned}$$

The cell-centered value of B_θ can be derived similarly, which is

$$\begin{aligned} B_\theta \quad = \quad & \frac{1}{V}(B_{\theta_2} A_{\theta_2}\frac{\cos\theta_1\Delta\theta + \sin\theta_1 - \sin\theta_2}{\cos\theta_1 - \cos\theta_2} + B_{\theta_1} A_{\theta_1}\frac{\sin\theta_2 - \sin\theta_1 - \cos\theta_2\Delta\theta}{\cos\theta_1 - \cos\theta_2}) + \\ & \frac{1}{V}(\Delta_\theta B_{r_1} r_1^2 - \Delta_\theta B_{r_2} r_2^2)\left(\tilde{\theta} - (\cos\theta_1\Delta\theta + \sin\theta_1 - \sin\theta_2)\theta_0\right), \end{aligned}$$

where

$$A_{\theta_1} = \frac{1}{2}(r_2^2 - r_1^2)\sin\theta_1,$$

$$A_{\theta_2} = \frac{1}{2}(r_2^2 - r_1^2)\sin\theta_2,$$

$$\tilde{\theta} = (\theta_1\cos\theta_1 - \sin\theta_1)\Delta\theta + \theta_1\sin\theta_1 + 2\cos\theta_1 - \theta_2\sin\theta_2 - 2\cos\theta_2,$$

$$\theta_0 = \frac{\theta_1 \cos\theta_1 - \theta_2 \cos\theta_2 + \sin\theta_2 - \sin\theta_1}{\cos\theta_1 - \cos\theta_2}$$

It is easy to see that the second term in equation of B_r is of $O((\Delta r)^2)$. Because

$$\int_{\theta_1}^{\theta_2} (c(\theta) - \theta_0)(\cos\theta_1 - \cos\theta) d\theta = \tilde{\theta} - (\cos\theta_1 \Delta\theta + \sin\theta_1 - \sin\theta_2)\theta_0$$

where $c(\theta) = (\theta_1 \cos\theta_1 - \theta\cos\theta + \sin\theta - \sin\theta_1)/(\cos\theta_1 - \cos\theta)$, the second term of B_θ is also of $O(\Delta\theta^2)$. Therefore, if only second order accuracy is needed, the second term can be neglected.

7. Numerical Experiments

In this section, we provided some examples to test our code. Without specification, the unsplit version of our solvers with CFL number of 0.6 is used. The time step is determined adaptively according to the current wave speed and CFL number. The computation is done on our Linux PC with 1.7GHz AMD Athlon processor. Whenever the parallel computation is needed, it is done on a four-processor cluster with 1.7GHz AMD Athlon processor.

7.1. Rotated shock-tube problem

This example has been used by (Tóth 2000) to compare several numerical schemes for MHD. We adopt the same initial and boundary conditions as (Tóth 2000). The initial left and right states are

$$(\rho, v_\parallel, v_\perp, p, B_\parallel, B_\perp) = \begin{cases} (1, 10, 0, 20, 5/\sqrt{4\pi}, 5/\sqrt{4\pi}), & \text{left,} \\ (1, -10, 0, 1, 5/\sqrt{4\pi}, 5/\sqrt{4\pi}), & \text{right.} \end{cases}$$

The v_z and B_z components are zero. It was proposed as a 1-D problem. However we solved it as a 2-D problem with an angle $\alpha = \tan^{-1} 2$ between the shock interface and y-axes. Initial domain and problem set-up is the same as in (Tóth 2000) except that the local spacing is different. We used 400 cells in x-direction. The number of cells in y-direction is equal to the number of ghost cells. In (Tóth 2000), $dy = dx$ was used, which results in that the shock interface is not in a straight line and the the parallel component of the magnetic field is not conserved as it should be even if the flux-CT is used. We modified the local spacing in y-direction so that the shock interface has a straight line. For $\alpha = \tan^{-1} 2$, we set $dy = dx/2$. After this modification, the parallel component of the magnetic field

$B_{\|} = 5/\sqrt{4\pi}$ is conserved exactly if the flux-CT is used. Fig. 7.1 shows the results of the parallel component of the magnetic fields for Roe's Riemann solver with different treatments of the $\nabla \cdot B$.

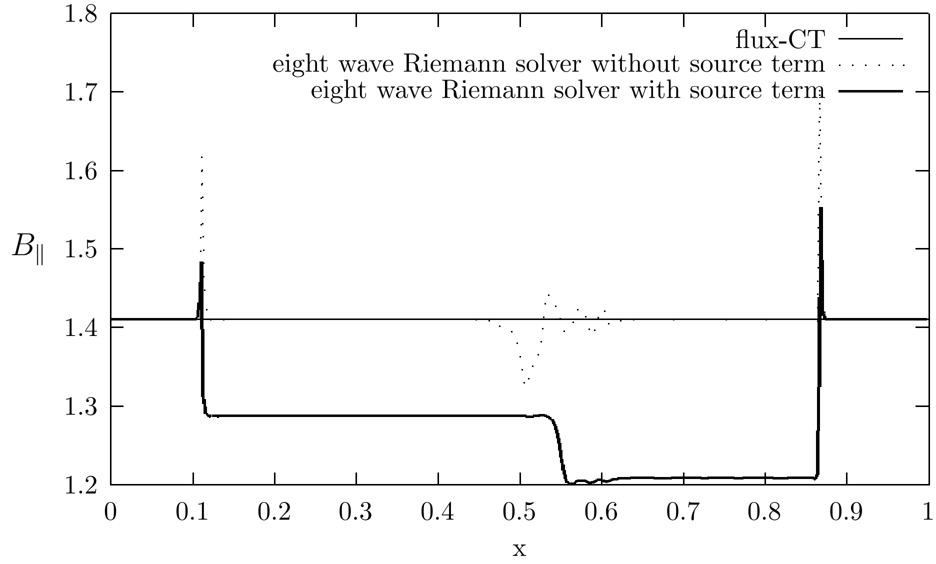

Fig. 7.1.— The parallel component of the magnetic field is shown for Roe's Riemann solver with different treatments of the $\nabla \cdot B$. The flux constraint transport (flux-CT) method preserves $B_{\|}$ exactly, The Powell's method with source term produces incorrect jump condition at the fast shock, which has been pointed out in (Tóth 2000).

We also tested this problem with other Riemann solvers. To achieve the second order accuracy, we used modified minmod limiter with parameter $\theta = 1.5$ in our reconstruction and two-stages second order Runge-Kutta method in time integration. For all of the Riemann solver and integration schemes, the parallel component of the magnetic field is preserved exactly when the flux-CT is used and has incorrect jump conditions for Powell's source term. We also compared other components of the solutions and the efficiency of different methods with flux-CT. The results are shown in Figs 7.2-a to 7.2-d.

We also tested different schemes with MUSCL-Hancock approach to achieve the second order accuracy. Compared with the two-stage Runge-Kutta method, MUSCL-Hancock is much faster. Figs 7.3-a to 7.3-d show the results of the HLLC Riemann solver with MUSCL-Hancock and two-stage Runge-Kutta approaches.

The Strang splitting version of the our solver was also tested. It seems that the CFL number can be larger for the splitting version than for the unsplit version. We have used a

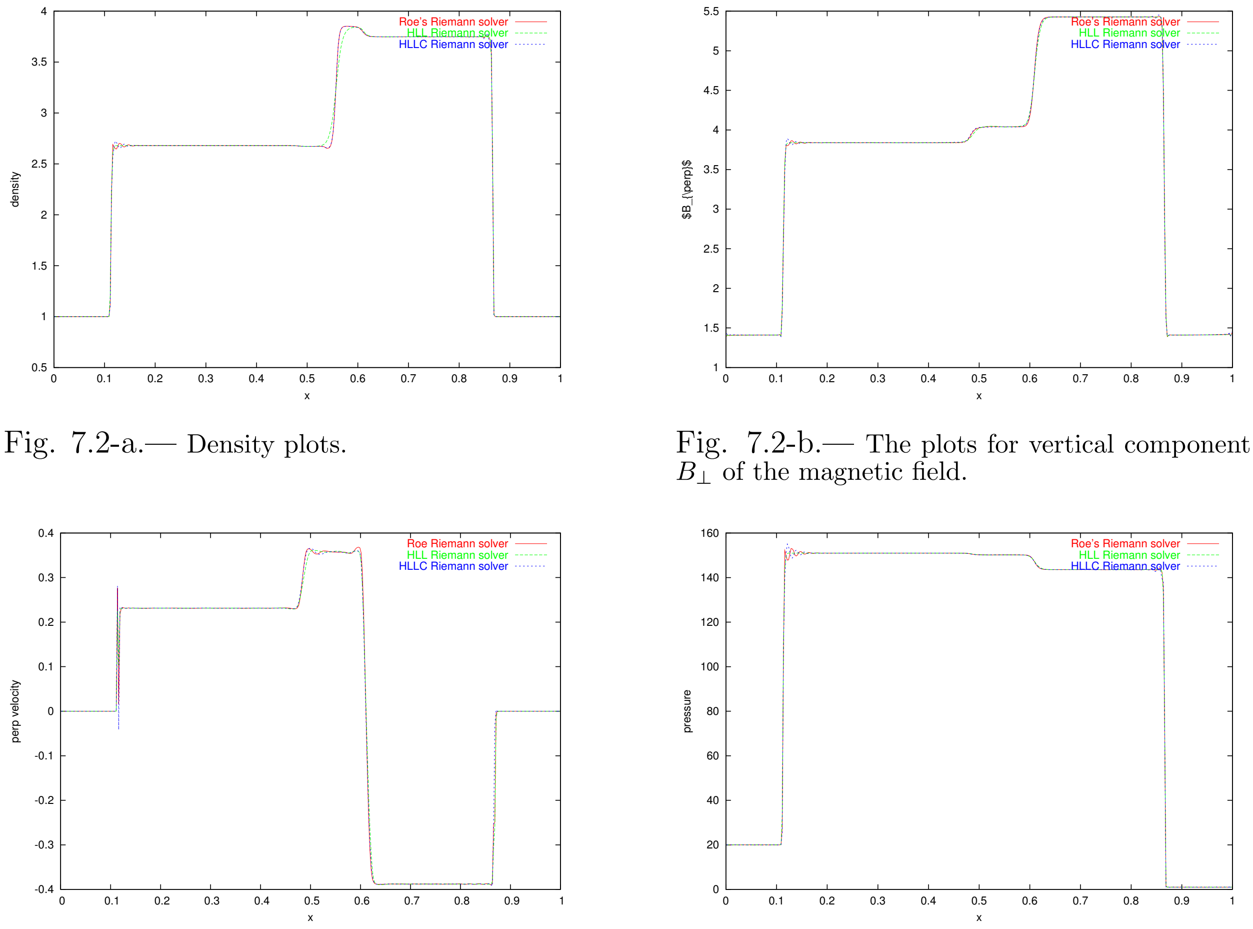

Fig. 7.2-a.— Density plots.

Fig. 7.2-b.— The plots for vertical component $B_{\perp}$ of the magnetic field.

Fig. 7.2-c.— The plots for vertical component $v_{\perp}$ of the velocity field.

Fig. 7.2-d.— Pressure plots.

Figure7.2: Results of different Riemann solvers with second order Runge-Kutta integration methods for the shock tube problem. The HLL Riemann solver has too much diffusion on the contact discontinuity and slow shock. The HLLC Riemann solver obtained almost the same results as Roe's Riemann solver. For CPU time, HLL took 7 seconds, HLLC took 8 seconds, and Roe's Riemann solver took 13 seconds. The oscillations in $B_{\perp}$ might be due to the geometry set-up. For angle of 45° and 0°, the oscillations disappear.

CFL number of 0.9 without any problem, whereas a CFL number of 0.7 for unsplit version caused instability for this problem. We think the signal speed considered in CFL condition for the unsplit version should be calculated as $v = \sqrt{(}v_x^2 + v_y^2)$ while for the split version, only v_x and/or v_y are considered. From the plots in Fig. 7.4, we can see that the numerical results are almost the same for both versions.

We also test this problem with the fifth-order WENO scheme combined with the third-order Runge-Kutta TVD method. The magnetic fields are advanced with flux-CT schemes.

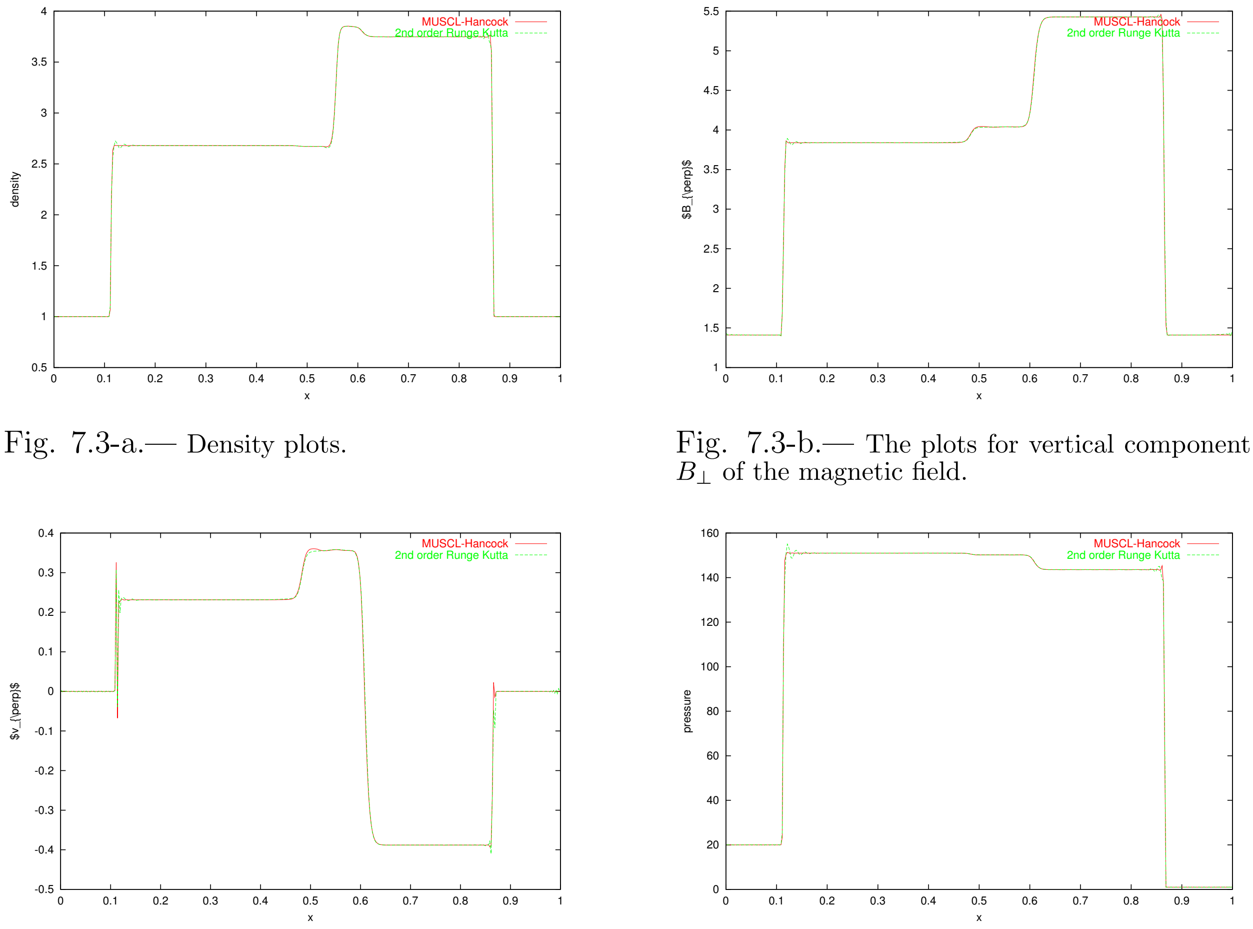

Fig. 7.3-a.— Density plots.

Fig. 7.3-b.— The plots for vertical component $B_{\perp}$ of the magnetic field.

Fig. 7.3-c.— The plots for vertical component $v_{\perp}$ of the velocity field.

Fig. 7.3-d.— Pressure plots.

Figure7.3: Results of HLLC Riemann solvers with MUSCL-Hancock approach, compared with the results for second order Runge-Kutta integration method for the shock tube problem. MUSCL-Hancock approach took only 5 seconds CPU time whereas the 2nd order Runge-Kutta approach took 8 seconds. The MUSCL-Hancock approach also reduces the oscillation near the fast shocks.

The results are very close to ‘those obtained from Roe's method combined with MUSCL-Hancock approach. However, it took a very long computation time, 179 seconds. One of the reasons is that it requires 3 ghost cells in each direction.

7.2. Two-dimensional propagation of Alfven wave

This example is taken from (Clarke 1996). It is illustrated in (Clarke 1996) that some schemes that perform quite well for hydrodynamical tests may have difficulties with the

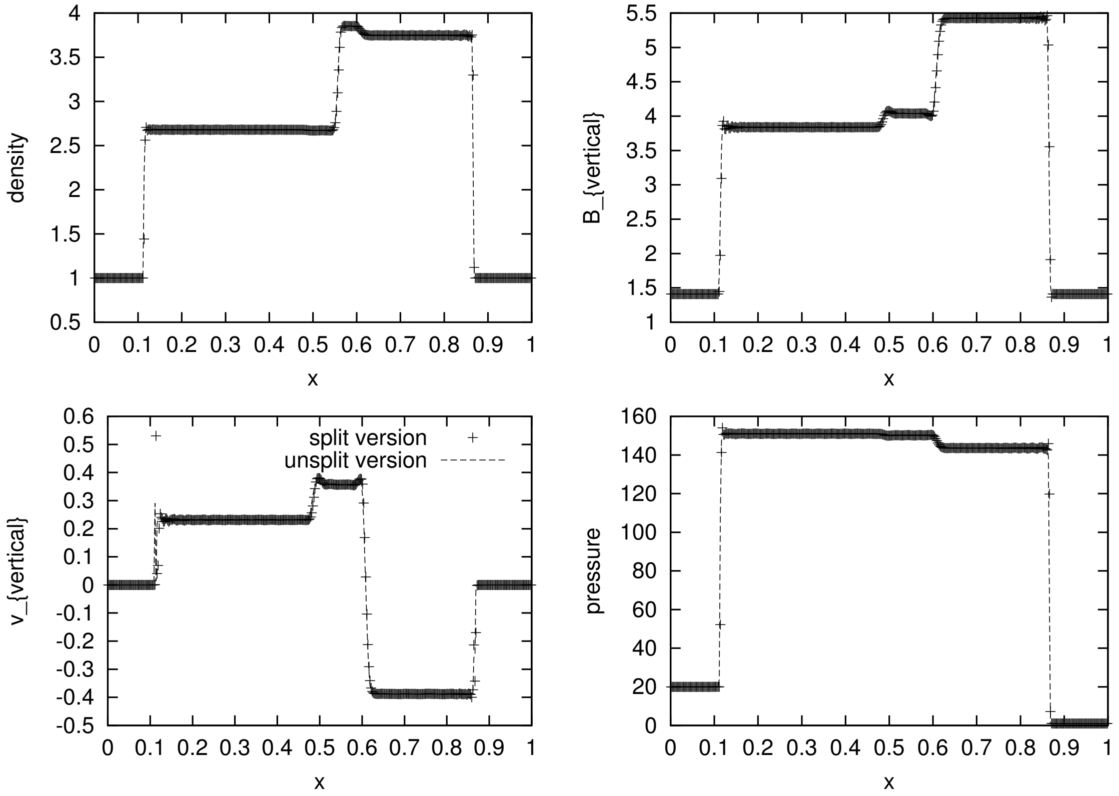

Fig. 7.4.— The comparison of the results for split and unsplit versions of Roe's method to rotated shock tube problem.

propagation of Alfven waves. The set-up of the problem is as follows. A circular pulse of velocity perpendicular to the plane of computation is initialized at the center of a 200×200 zone grid which contains a uniform magnetic field. Throughout the (1,2)-plane, the density, pressure, and adiabatic index are set to 1, 3/5, and 5/3, respectively. The velocity is set to zero everywhere, except for a circular region in the center of the grid with a radius if 10 zones in which v_3 is set to 10^{-3}. In (Clarke 1996), the Alfven pulse has been either transported in x-direction or in diagonal direction. We also tested the problem with a different propagation direction by setting the magnetic field is $B_x = 1, B_y = 2$. This problem has exact solutions for v_3. The circular pulse in v_3 should be carried along the magnetic field line at the Alfven speed intact and undistorted.

Figs. 7.5-a to 7.5-d show the results of Roe's Riemann solver combined with the Hancock-MUSCL time integration.

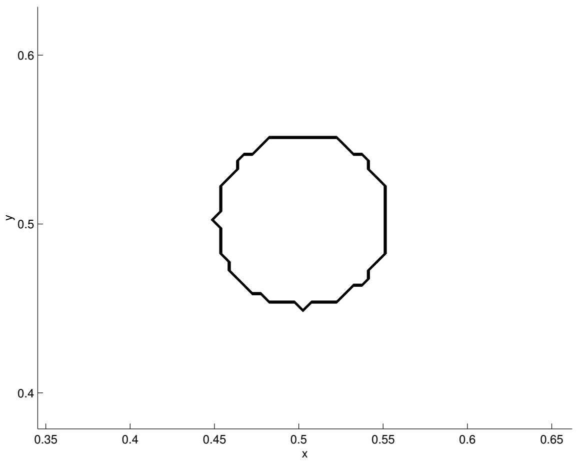

Fig. 7.5-a.— Contours of velocity perpendicular to the grid (v_3) at $t = 0$.

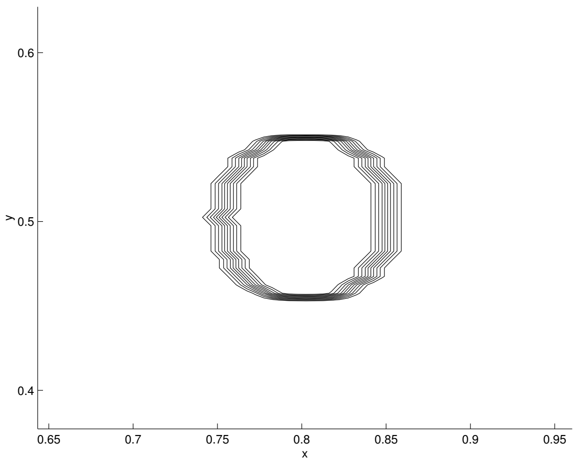

Fig. 7.5-b.— Contours of velocity perpendicular to the grid (v_3) at $t = 0.3$. $B_x = 1, B_y = 0$.

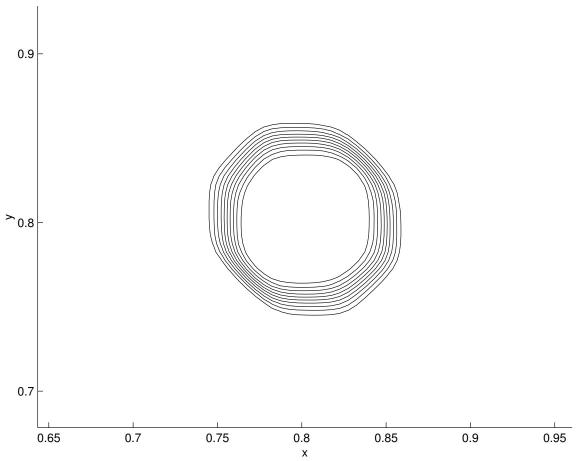

Fig. 7.5-c.— Contours of velocity perpendicular to the grid (v_3) at $t = 0.3$. $B_x = 1, B_y = 1$.

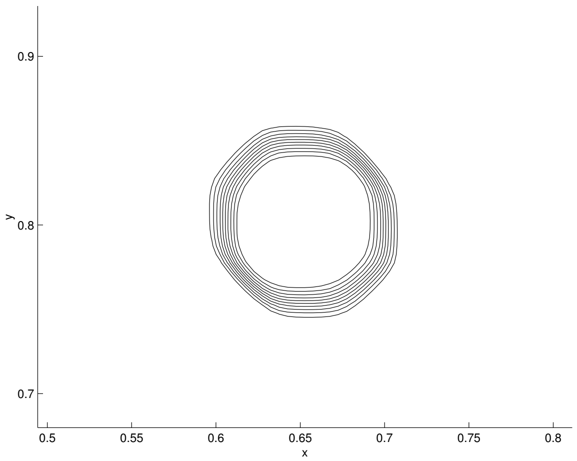

Fig. 7.5-d.— Contours of velocity perpendicular to the grid (v_3) at $t = 0.15$. $B_x = 1, B_y = 2$.

We also solved this example with other Riemann solvers. The solutions for the HLL family solvers are very bad when CFL=0.6 was used. However the results for CFL=0.5 are relatively good. Figs. 7.6-a and 7.6-b show the results of HLLC and HLL Riemann solver for CFL=0.5. It is clear that HLL Riemann solver is more diffusive than HLLC Riemann solver. The splitting version of our solvers works well for this problem with CFL=0.8.

7.3. Dai and Woodward's cloud-shock interaction

This test problem is taken from (Dai & Woodward 1998) and was also used by (Tóth 2000) to compare different schemes. It models the disruption of a high density cloud by a

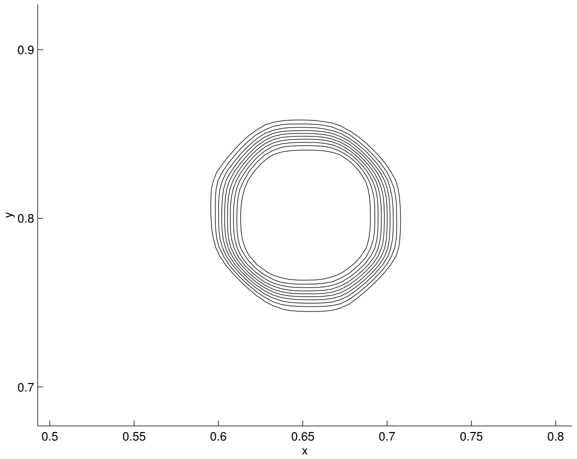

Fig. 7.6-a.— Contours of velocity perpendicular to the grid (v_3) at $t = 0.15$. $B_x = 1, B_y = 2$. HLLC Riemann solver is used.

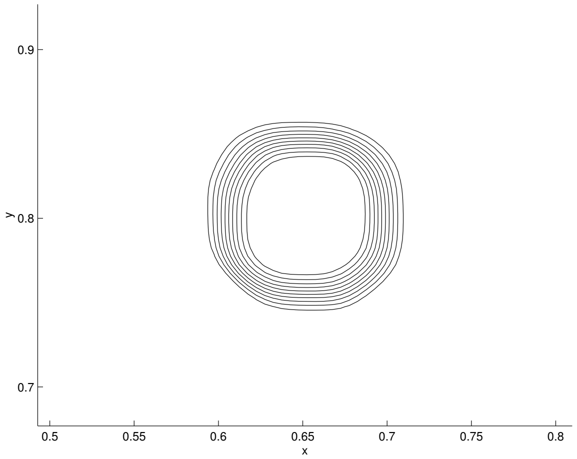

Fig. 7.6-b.— Contours of velocity perpendicular to the grid (v_3) at $t = 0.15$. $B_x = 1, B_y = 2$. HLL Riemann solver is used.

strong shock wave.

We solved this example using our AMR schemes. AMR with two-level and three-level refinement is tested with Roe's solver and HLLC Riemann solver. The default refinement ratio is 2 without specification. The second order accuracy is achieved via MUSCL-Hancock approach. The divergence-free condition of the magnetic field is preserved very well (around round of errors) by the flux-CT schemes. Roe's solver took about 45 seconds for two-level AMR and 159 seconds for three-level AMR. HLLC solver took about 32 seconds for two-level AMR and 116 seconds for three level AMR. The results of contour plots of density and $\nabla \cdot B$ are shown in Figs. 7.7-a to 7.8-d.

The WENO scheme has difficulty in solving this problem. It fails with using the CFL=0.6 and works only when CFL<0.3.

We also tested our AMR schemes with different refinement ratios. Figs. 7.9-a and 7.9-b show the results of three-level refinement where the refinement ratio is 3 between the first two levels and 2 between the last two levels.

It was found in (Tóth 2000) that the dimensional split one step TVD scheme can easily fail due to non-physical states produced during the violent collision of the shock and the cloud even when the minmod limiter and an entropy fix were used. However, we did not experience such problems in our test with any of our Riemann solvers for CFL=0.9. We calculated with our parallel AMR version. The results for Roe's Riemann solver is shown in Figs.7.10. The speed-up for the four-processor parallel compuatation is 2.2 with two-level

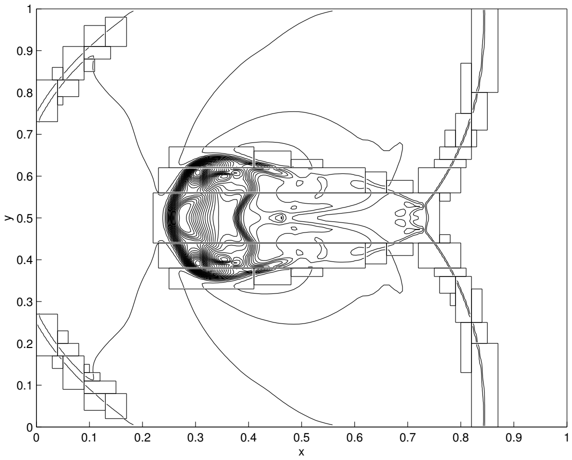

Fig. 7.7-a.— Density contour plot for Roe's method with two-level refinement

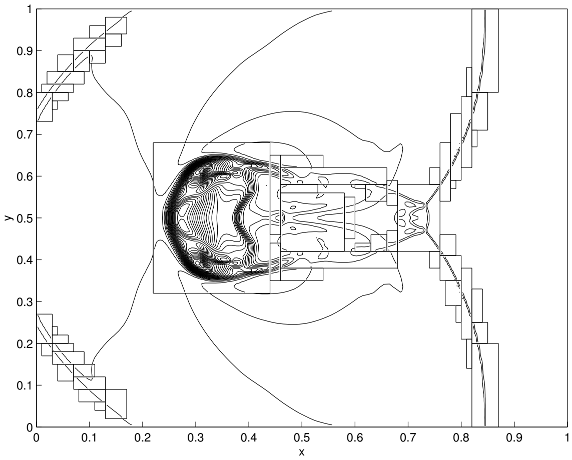

Fig. 7.7-b.— Density contour plot for HLLC solver with two-level refinement.

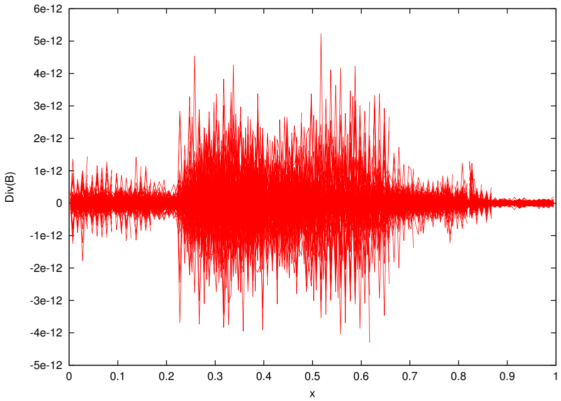

Fig. 7.7-c.— $\nabla \cdot B$ plot for Roe's method with two-level refinement.

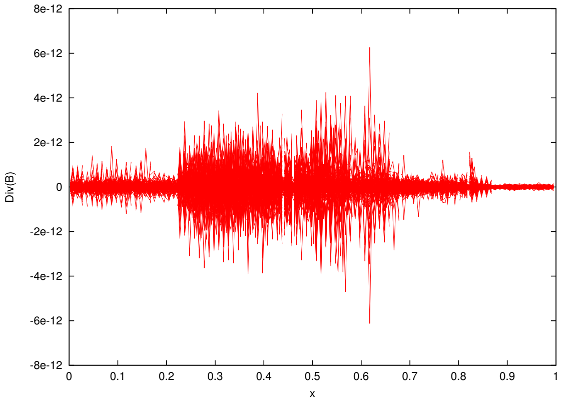

Fig. 7.7-d.— $\nabla \cdot B$ plot for HLLC solver with two-level refinement.

refinement and is 3.5 without refinement. It was found that the parallel efficiency beccame smaller with the increase of the refinement level.

This test example has symmetry along the line $y = 0.5$. Therefore, we can solve the problem with a half domain [0,1]×[0,0.5], providing symmetric boundary conditions at $y = 0.5$. The result of HLLC with two refinement levels is shown in Fig. 7.10.

7.4. Balsara's Rotor problem

This test problem is taken from (Balsara & Spicer 1999). It was also used by (Tóth 2000) to compare several numerical schemes. We used exactly the same set-up of the problem as was described in (Tóth 2000).

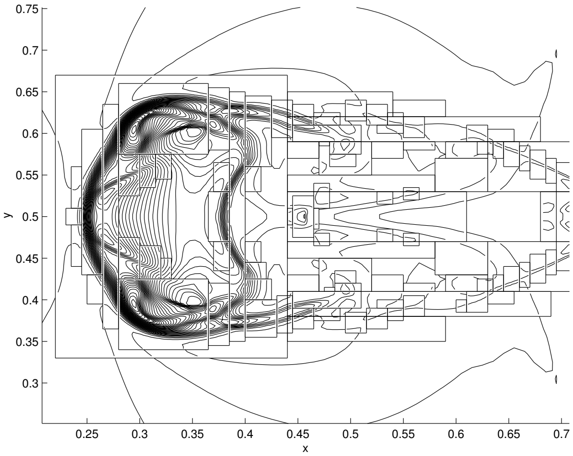

Fig. 7.8-a.— Density contour plot for Roe's method with three-level refinement.

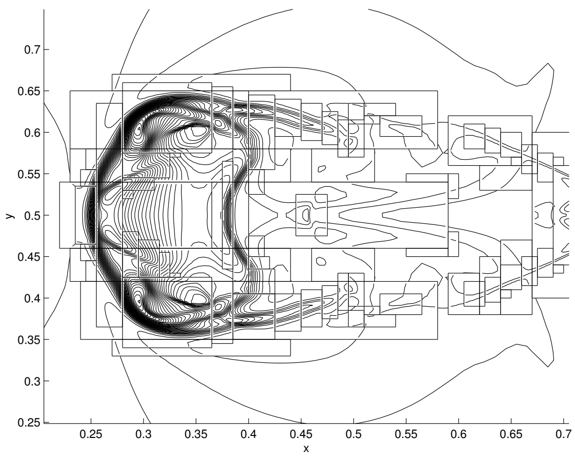

Fig. 7.8-b.— Density contour plot for HLLC solver with three-level refinement.

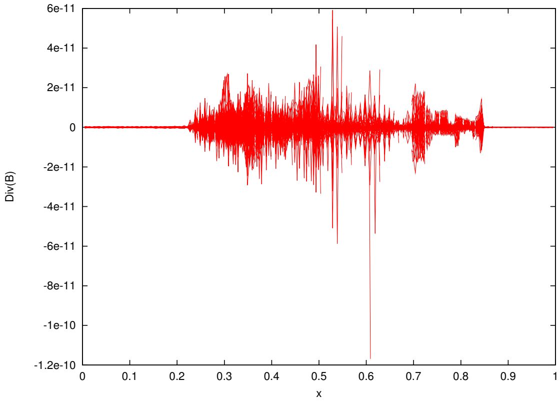

Fig. 7.8-c.— $\nabla \cdot B$ plot for Roe's method with three-level refinement.

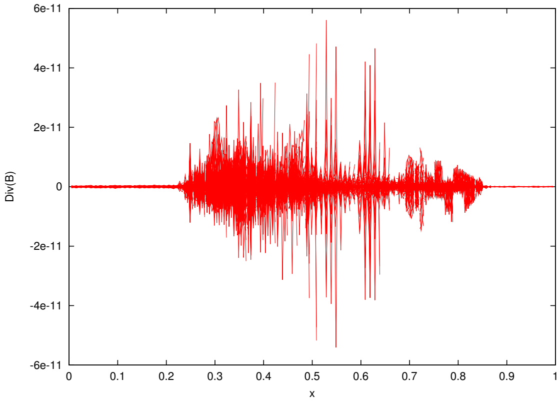

Fig. 7.8-d.— $\nabla \cdot B$ plot for HLLC solver with three-level refinement.

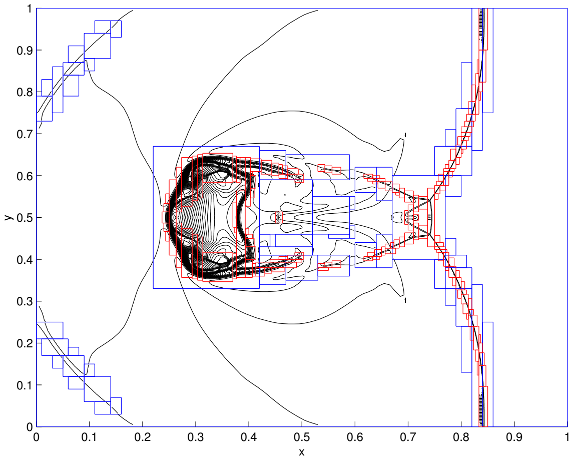

Fig. 7.9-a.— Density contour plot for HLLC solver with three-level refinement. The refinement ratio is 3 between the first two levels.

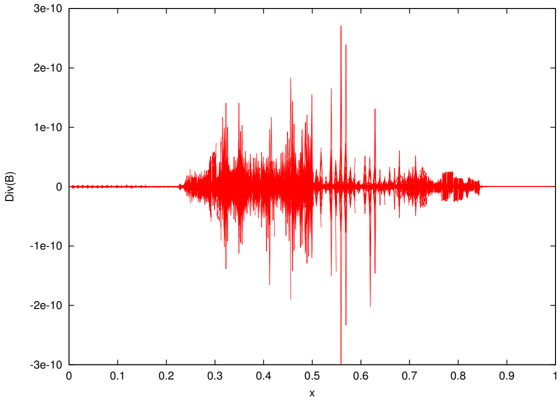

Fig. 7.9-b.— $\nabla \cdot B$ plot for HLLC solver with three-level refinement. The refinement ratio is 3 between the first two levels.

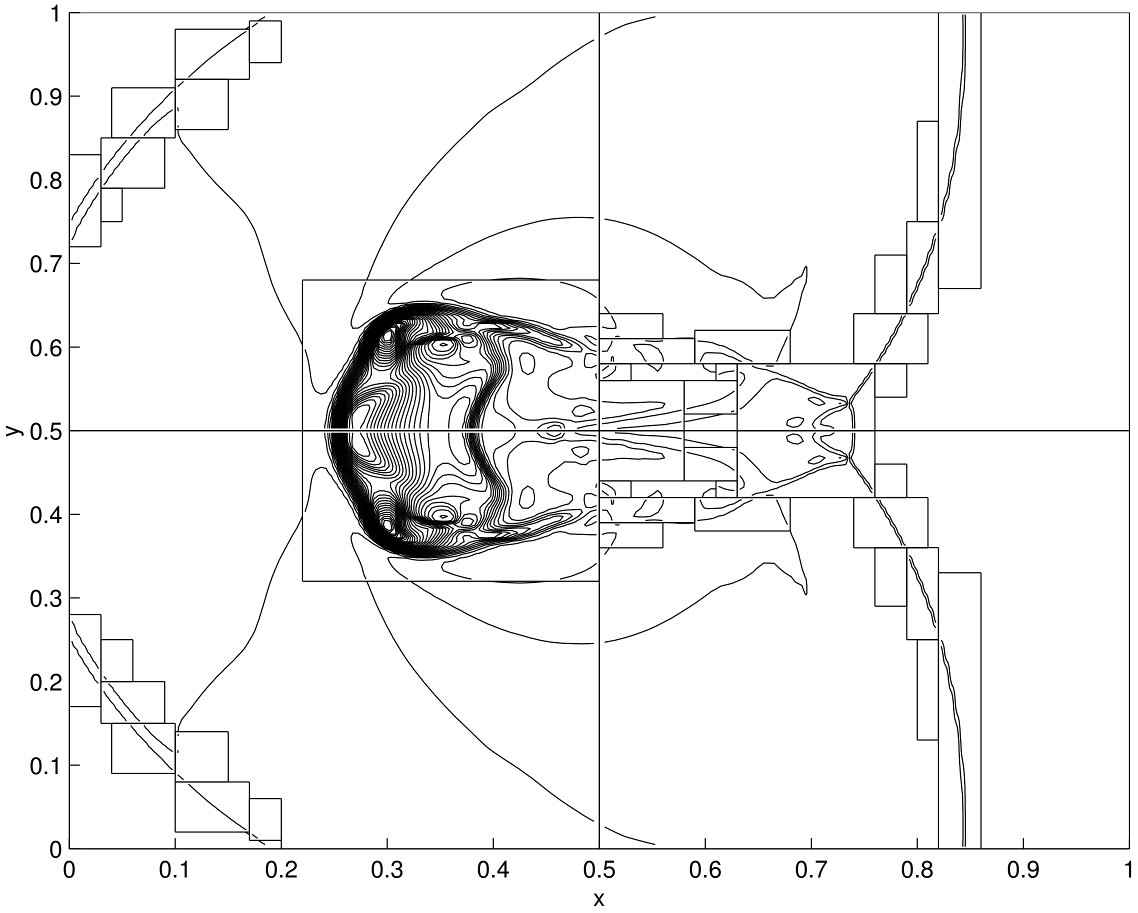

Fig. 7.10.— The density contour plot for shock-cloud interaction problem for dimensional split Roe's method. The refinement ratio is 2. 4 processors were used. The base grid is 100×100.

We first solved the first rotor problem of (Tóth 2000) to time 0.15. It was reported in (Tóth 2000) that many one step TVD base scheme failed to solve this problem due to negative pressure. We did not encounter any difficulties with all of our solvers and time integration schemes. We used two-level refinement with refinement ratio of 3. Figs. 7.11-a to 7.11-f show the results for Roe's, HLLC and HLL Riemann solver combined with MUSCL-Hancock approach. HLL solver took 34 seconds, HLLC solver took 37 seconds, and Roe's solver took 52 seconds to reach the final time $t = 0.15$. It is clearly seen that the HLL solver was too diffusive.

The dimensional unsplit TVD-MUSCL scheme was suggested to solve this problem by (Tóth 2000). However, our dimensional split solvers solve this example without any difficulty. We used a large CFL number 0.9 and two-level refinement with refinement ratio of 3. The

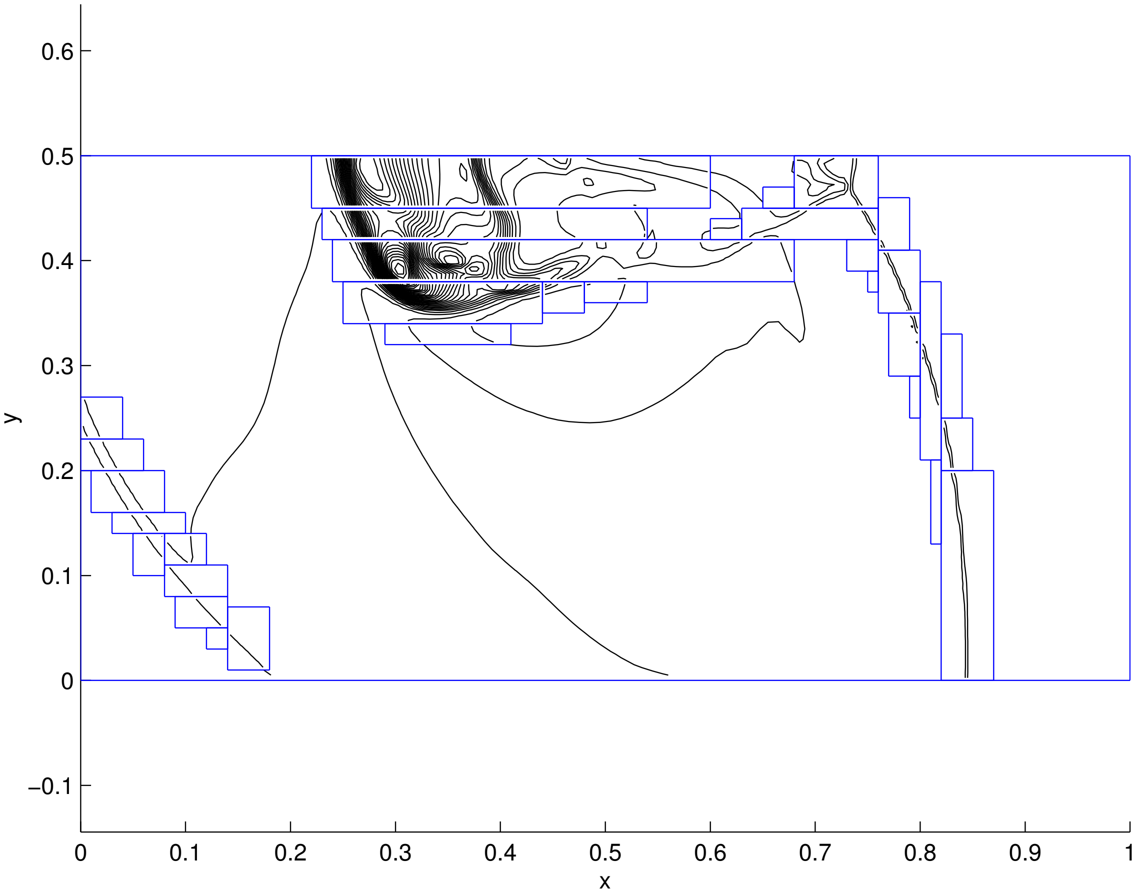

Fig. 7.10.— The density contour plot for shock-cloud interaction problem when it is solved with half domain and reflecting boundary at the top.

results for different Riemann solvers are shown in Fig.7.12. Again we tested the parallel efficiency and found the efficiency was 90% without refinement and 70% with two-level refinement.

The second rotor problem, described in (Tóth 2000), was also solved by our codes. We used a three-level refinement with refinement ratio of 2 and base grid 100×100. The results are in excellent agreement with those plotted in (Tóth 2000). We also solved the second rotor problem (Tóth 2000) to the time $t = 0.295$. Figs. 7.13-a t0 7.13-b show the results of HLLC solver with MUSCL-Hancock approach.

As a test to our code, we also solved the second motor problem with cylindrical coordinates. It is important to correctly initialize the magnetic field in (r, θ) coordinates. It turns

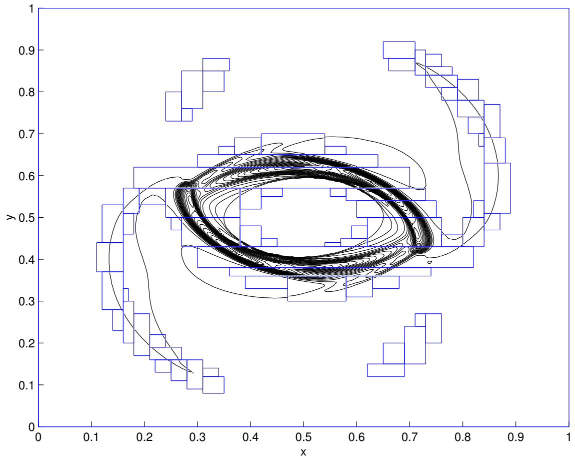

Fig. 7.11-a.— Density contour for Roe's solver. 30 contours between 0.483 and 12.95 are used.

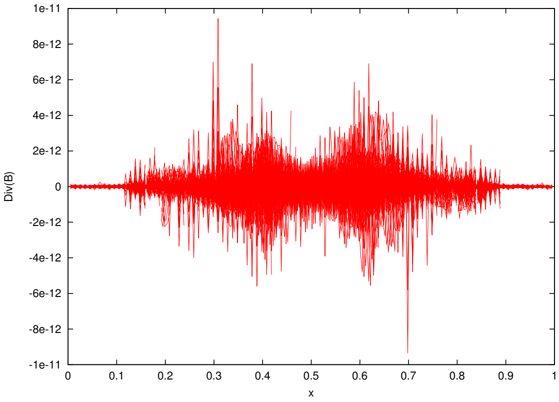

Fig. 7.11-b.— $\nabla \cdot B$ for Roe's solver

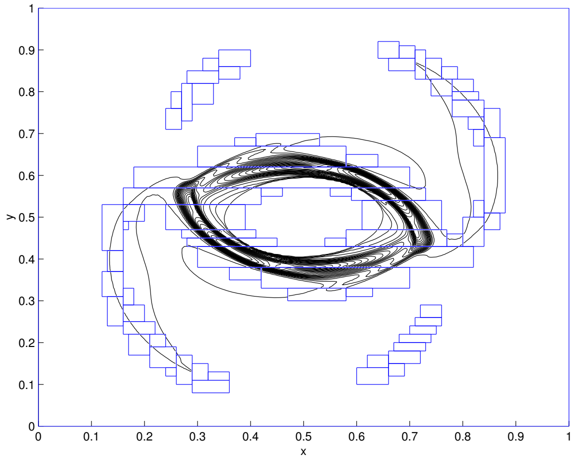

Fig. 7.11-c.— Density contour for HLLC solver.

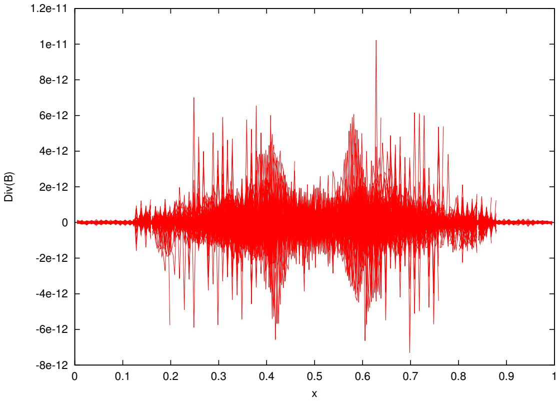

Fig. 7.11-d.— $\nabla \cdot B$ for HLLC solver.

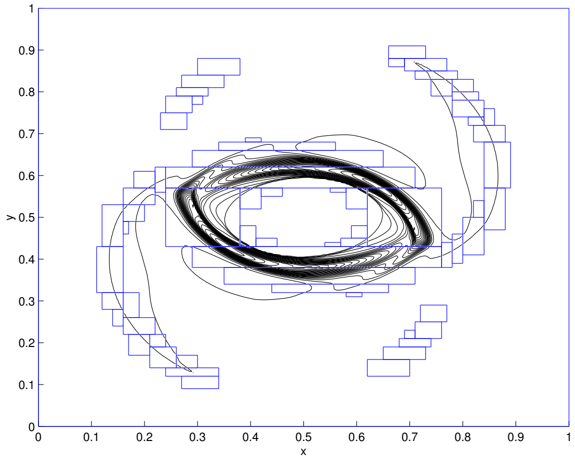

Fig. 7.11-e.— Density contour for HLL solver.

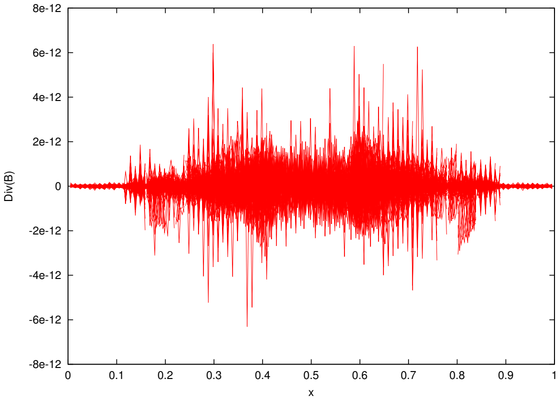

Fig. 7.11-f.— $\nabla \cdot B$ for HLL solver.

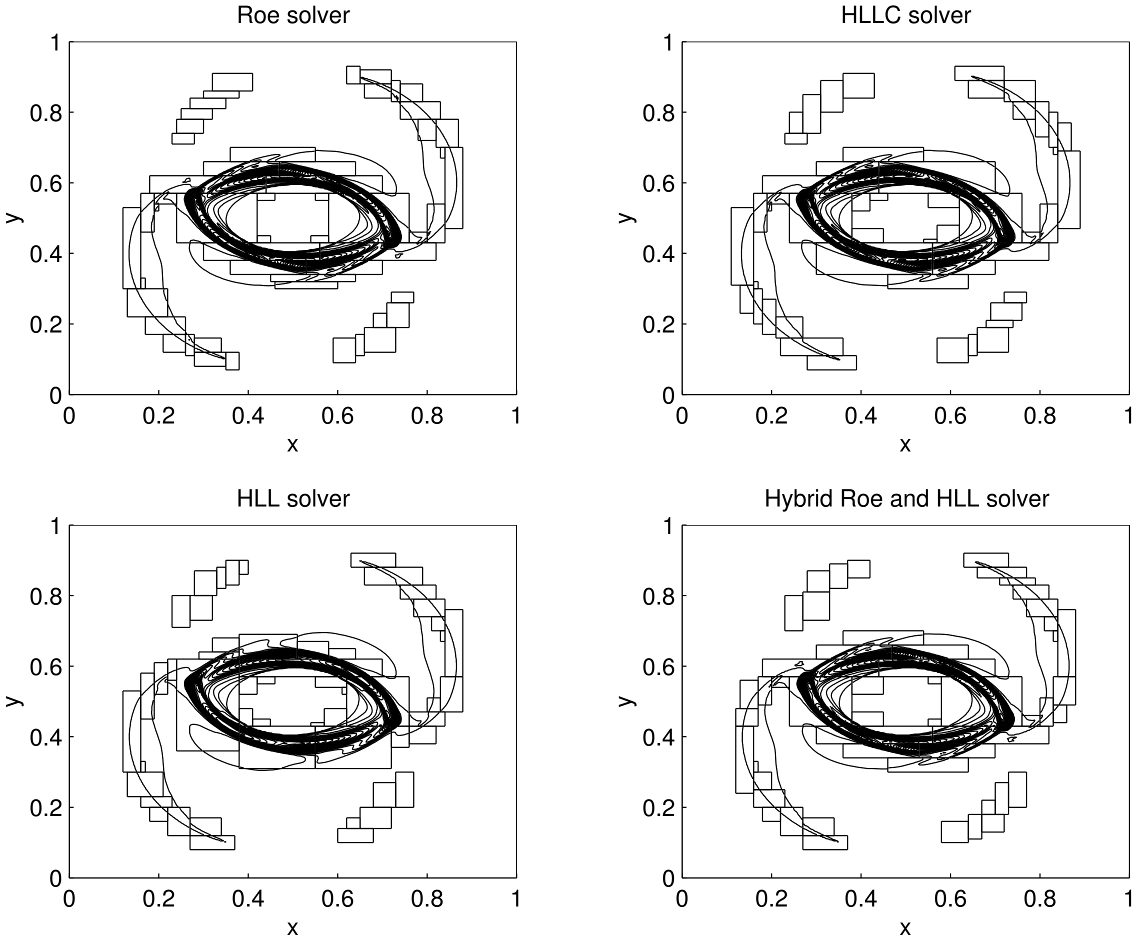

Fig. 7.12.— Results for rotor problems with dimensional split solvers . 30 contours between 0.483 and 12.95 are used.

out that the exact solution

$$\begin{aligned} B_r &= B_x \cos(\theta), \\ B_\theta &= -B_x \sin(\theta), \end{aligned} \tag{35}$$

is not divergence-free in our finite-volume discretization. Therefore, we adopted an approximate initialization for B_r, which is

$$B_r = B_x \frac{\sin\theta_{j+1} - \sin\theta_j}{d\theta}.$$

This problem has two challenges for preserving the divergence-free condition on an AMR grid. The first is the singular point at the origin. B_r is not unique at origin. Although it

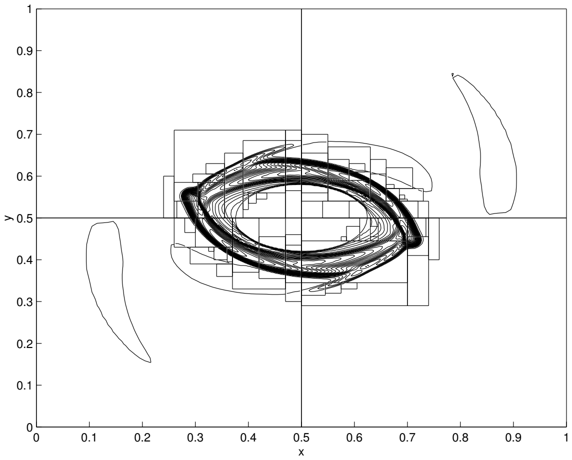

Fig. 7.13-a.— Density contours for the second rotor problem. 30 contour lines between 0.532 and 10.83 were used.

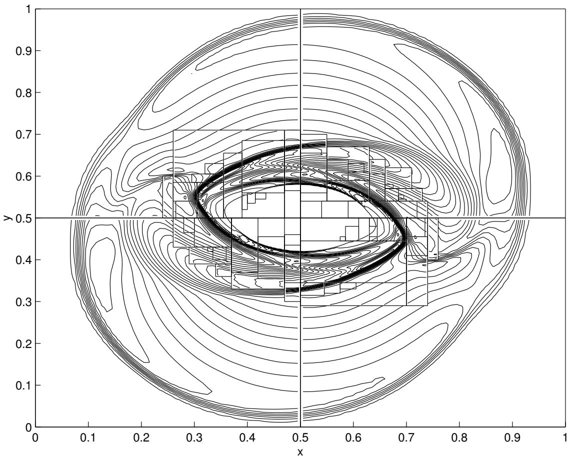

Fig. 7.13-b.— Pressure contours for the second rotor problem. 30 contours betwen 0.007 and 0.776 were used.

is not used in our finite-volume discretization of $\nabla \cdot B$ (canceled due to the zero area at $r = 0$), it is needed in calculating the cell-centered values of B_r, which is then used in the Riemann solver. To calculate the $\frac{dB_r}{dt}$ at the origin, we used the cell-centered values of $\frac{dB_r}{dt}$ at $(\theta, \frac{1}{2}dr)$, which is calculated by our Riemann solver, and the values calculated at $r_1 = dr$ in flux constraint-transport method.

The Riemann solver produces different electric-field values for different θ at origin. To maintain the divergence-free condition, only one electric-field value at the origin should be used to advance the B_θ for all of the θs. We set the electric-field at the origin to be the average over the whole circle.

The next challenge is the periodic boundary condition in θ direction. Every patch that shares an edge with $\theta = 0$ or $\theta = 2\pi$ can become potentially an electric-field correction partner for a coarse patch on the other end. It is important to make sure that the correction does only once for each cell, and the B_θ and the electric-field have the same values (up to the round-off error) at $\theta = 0$ and $\theta = 2\pi$. The density contour plots with refinement and values of $\nabla \cdot B$ over the whole domain are shown in Figs. 7.14-a and 7.14-b.

7.5. Rayleigh-Taylor instability

We also tested our code to solve the Rayleigh-Taylor instability problems. The computational set-up is as follows. The initial domain is [0,0.25]×[0,1]. The initial densities of

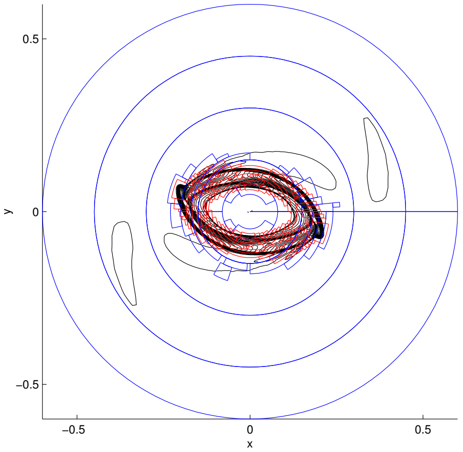

Fig. 7.14-a.— Density contour plots with three-level refinement for rotor problem. $t = 0.295$. 30 contour-lines from 0.45 to 13 are used.

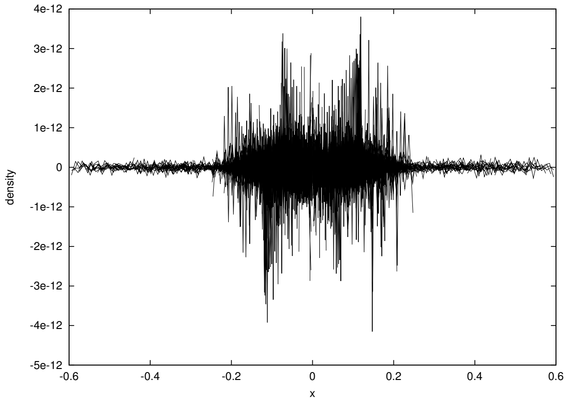

Fig. 7.14-b.— The $\nabla \cdot B$ at $t = 0.295$ for Fig. 7.14-a.

the heavy and light fluids are $\rho_1 = 1$ and $\rho_2 = 2$. Pressure equilibrium is assumed across the interface at $y = 0.5$. We used reflecting boundary conditions in both x and y directions. The detail of the problem is described in (Remachle et al. 2003). In (Remachle et al. 2003), the perturbation is applied to the velocity field, In our implementation, we applied the perturbation to the interface at $y = 0.5$. The initial interface is defined as

$$y = 0.01 \cos(8\pi x).$$

It was pointed out in (Jun & Norman 1995) that the critical strength of a magnetic field to suppress the instability of a mode of wavelength λ is

$$B_c = \frac{\sqrt{g\lambda(\rho_2 - \rho_1)}}{\cos\theta},$$

where g is the gravity acceleration, θ is the angle between $\mathbf{B}$ and the wave propagation direction. For our specification of the problem, $B_c = \sqrt{0.25} = 0.5$.

We tested this problem with two-level refinement with refinement ratio of 3. The local spacing for base grid is $\Delta x = \Delta y = 0.005$. We first tested the impact of the tangential magnetic field B_x on the growth of the instability. The final time is $t = 2.0$. We used $B_y = 0$ and different values of B_x: 0, $0.2B_c$, $0.5B_c$, and $0.8B_c$. The results are shown in Fig. 7.15. It is clearly seen that the instability is decreased dramatically with the increase of the B_x.

We also tested the impact of the normal magnetic field with choice of $B_x = 0$ and $B_y = B_c$. Unlike the tangential magnetic field, the normal magnetic field has less impact

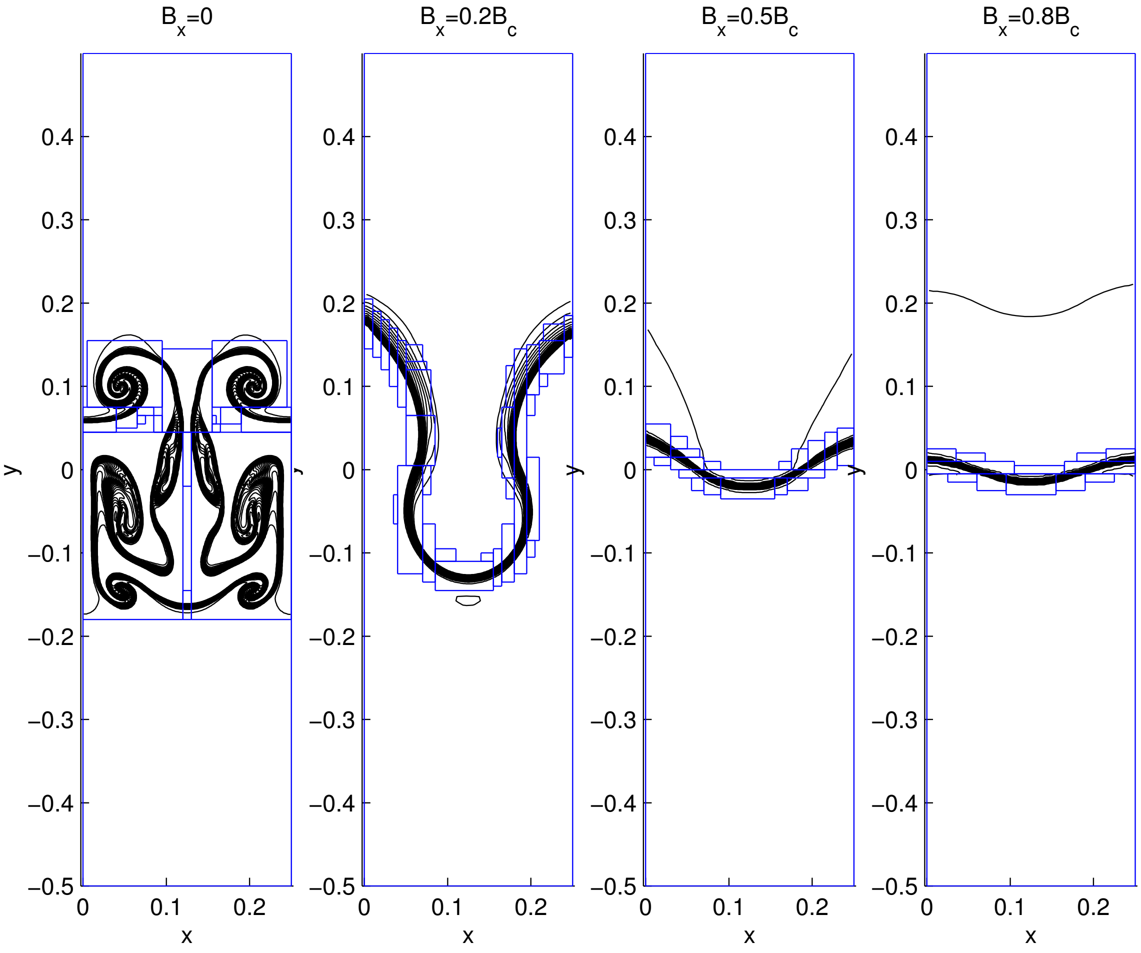

Fig. 7.15.— The impact of the tangential magnetic field to the growth of the instability. 30 density contour between 1 and 2 are used.

on the growth of the instability. Fig. 7.16 shows even applying a larger B_y than critical field strength cannot suppress the growth of in normal direction. However, the growth in tangential direction is suppressed owing to the strong tangential magnetic filed generated during the integration.

7.6. Magnetized jet problem

This example is used to test our code on the cylindrical geometry with (r, z) coordinates. It is introduced by (Ryu 2001). This is a simulation of a light cylindrical MHD jet with a top-hat velocity profile. We tested this problem with computation domain [0,1]×[0,2]. The

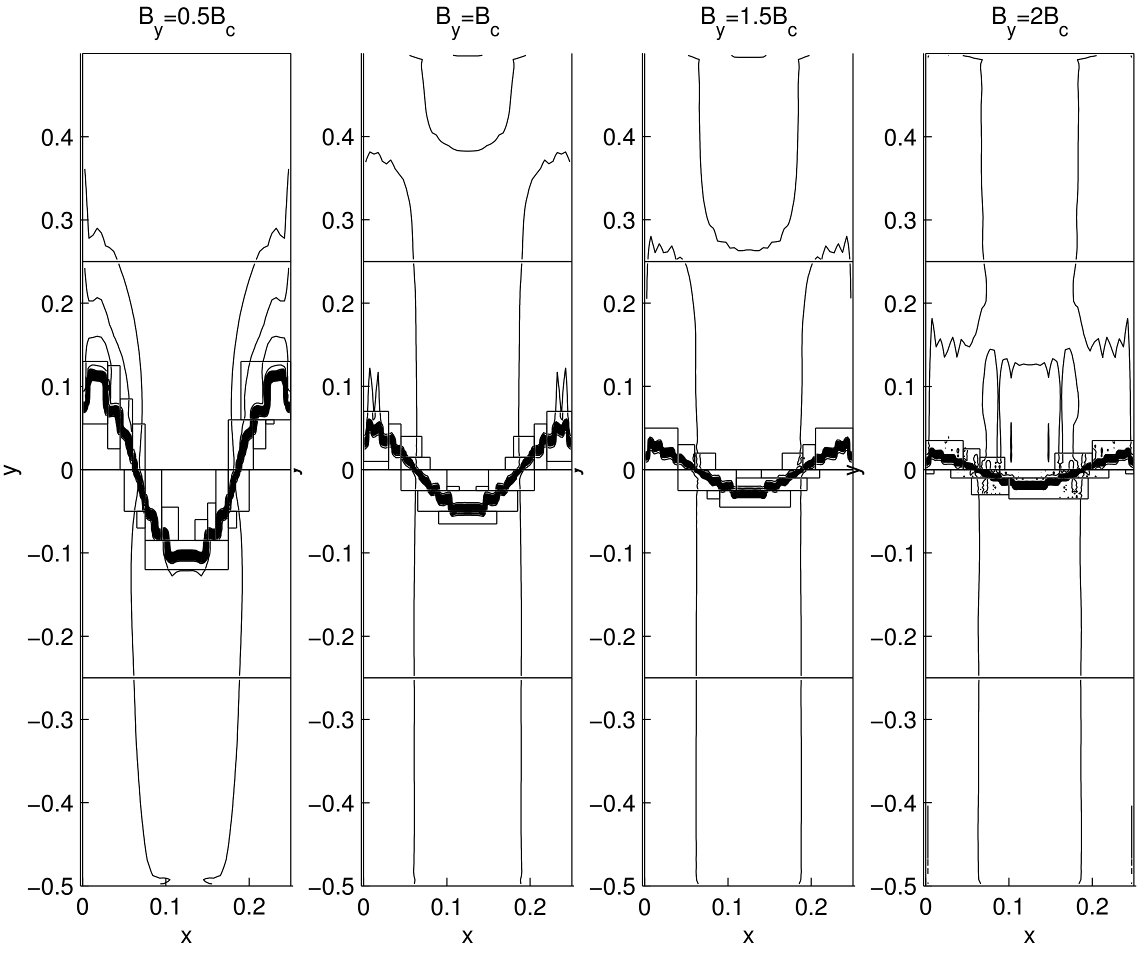

Fig. 7.16.— The impact of the normal magnetic field to the growth of the instability. 30 density contour between 1 and 2 are used.

base grid is 200×200. The jet has a radius 0.125, which is about 25 base grid cells. The ambient medium has sound speed of 1, and poloidal magnetic field ($B_\phi = B_r = 0$, $B_z = 0.1$). The jet has Mach number of 20, gas density contrast $\rho_{jet}/\rho_{\rm ambient} = 0.1$. The jet carries a helical magnetic field with $B_r = 0, B_\phi = 2B_{\rm ambient}(r/r_{jet})$, and $B_z = B_{\rm ambient}$.

We used three refinement levels with refinement ratio 3 for the first two levels and 2 for the last two levels. We ran our test until $t = 0.1$. Fig. 7.17-a shows the density contour plot with the refinement. Fig. 7.17-b shows the results of $\nabla \cdot \mathbf{B}$.

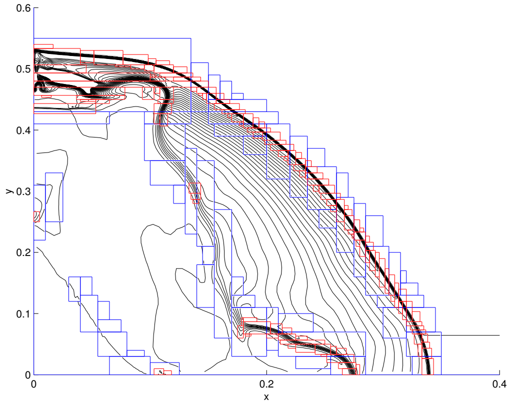

Fig. 7.17-a.— The "zoomed" version of refinement and density contour plot for the jet problem on cylindrical (r, z) plane.

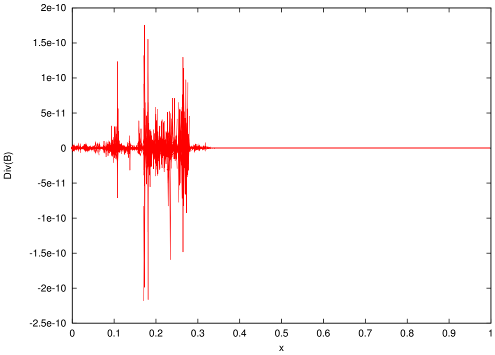

Fig. 7.17-b.— The $\nabla \cdot \mathbf{B}$ for Fig. 7.17-a.

7.7. Hot bubble problem

We used this example to test our spherical HD and MHD solver. The problem is set up as following. The computational domain is a spherical domain in (r, θ) with size $[0.2, 3.8] \times [0, \pi/3]$. The initial gas subjected to the external gravity is in equilibrium state with density and pressure $\rho = p = \exp(-r)$. The spherical bubble located at center (1.1,0) with radius 0.5. The density inside the bubble is defined as $\rho = 0.1 \exp(-r)$. We solved it with two-level refinement. The initial base grid is 360×120. The refinement ratio is 3. HLLC Riemann solver is used. We first solved the problem with no magnetic field, i.e., $B_r = B_\theta = B_\phi = 0$. Then we used a uniform magnetic field at large r in $z = r\cos\theta$ direction. It is a potential field with magnitude $F = -b \cdot r\cos\theta$. The magnetic field is defined by

$$B = -\nabla F = (b\cos\theta, -b\sin\theta).$$

The constant b controls the magnitude the magnetic field. We tested the bubble problem with $b = 0.4$.

The solutions without magnetic field are shown in Fig. 7.18. The Rayleigh-Taylor instability at the contact interface is clearly seen from the plot. The solutions with magnetic field are shown in Figs. 7.19 and 7.20. The Rayleigh-Taylor instability at the contact interface is suppressed by a strong magnetic field in Fig. 7.19.

It is very important to maintain the divergence-free condition of the magnetic field for this example. Otherwise, the pressure can easily become negative. The flux-CT approach combined with our AMR divergence-preserving interpolation preserved the divergence-free

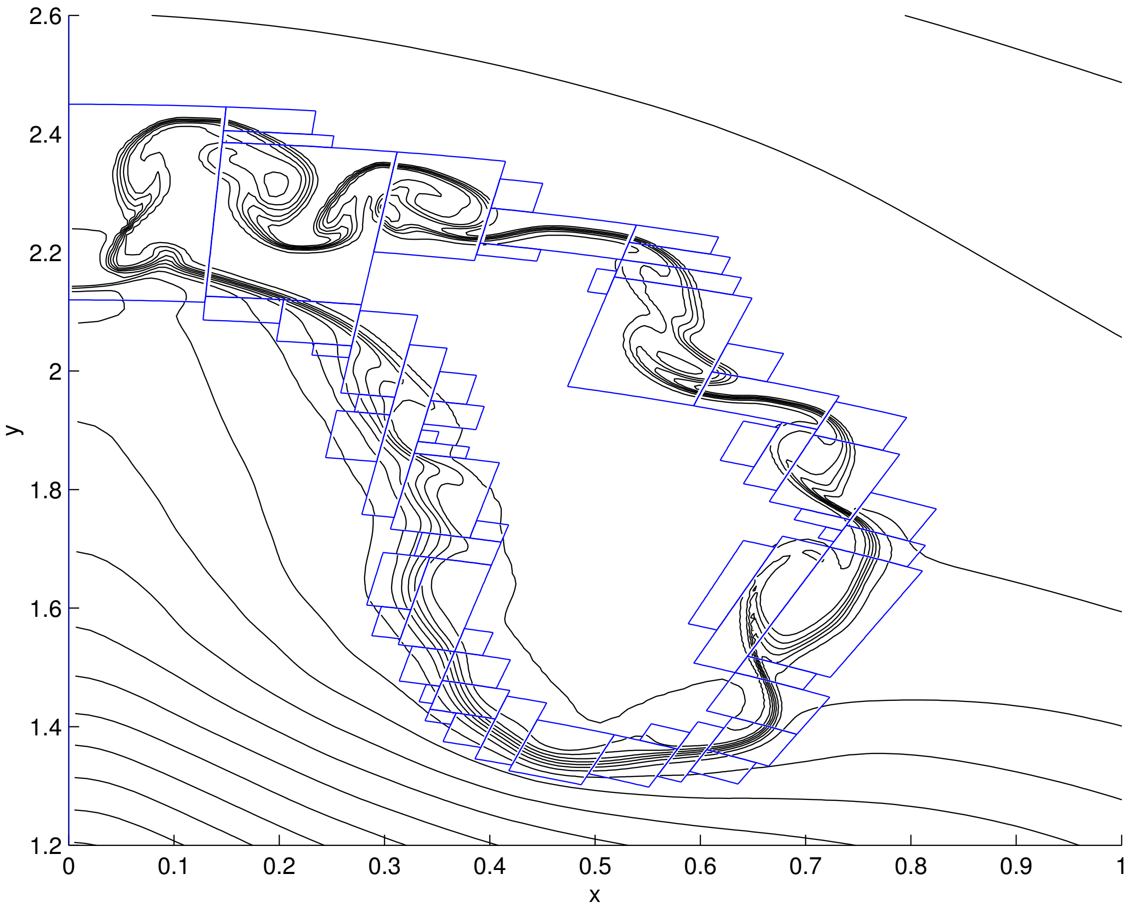

Fig. 7.18.— The “hot bubble” problem solved without magnetic field. 50 density contour between 0 and 1 are used.

condition very well. The divergence of the magnetic field versus $y = r\cos\theta$ is plotted in Fig.7.21.

We also tested this problem with other Riemann solvers and time integration schemes. The results for HLL solver is quite diffusive, while the results for Roe’s solver are very close to what we have shown. The HLLC took about 1254 to reach the time $t = 1.5$, while the Roe’s Riemann solver took about 1673 seconds. The same CFL number 0.6 is used for all of the solvers. We did not test the dimensional split version of our code for this problem. If that would be used, the different boundary conditions must be applied to each split step to maintain the HSE. The method similar to that described in (Zingle et al. 2002) should be used.

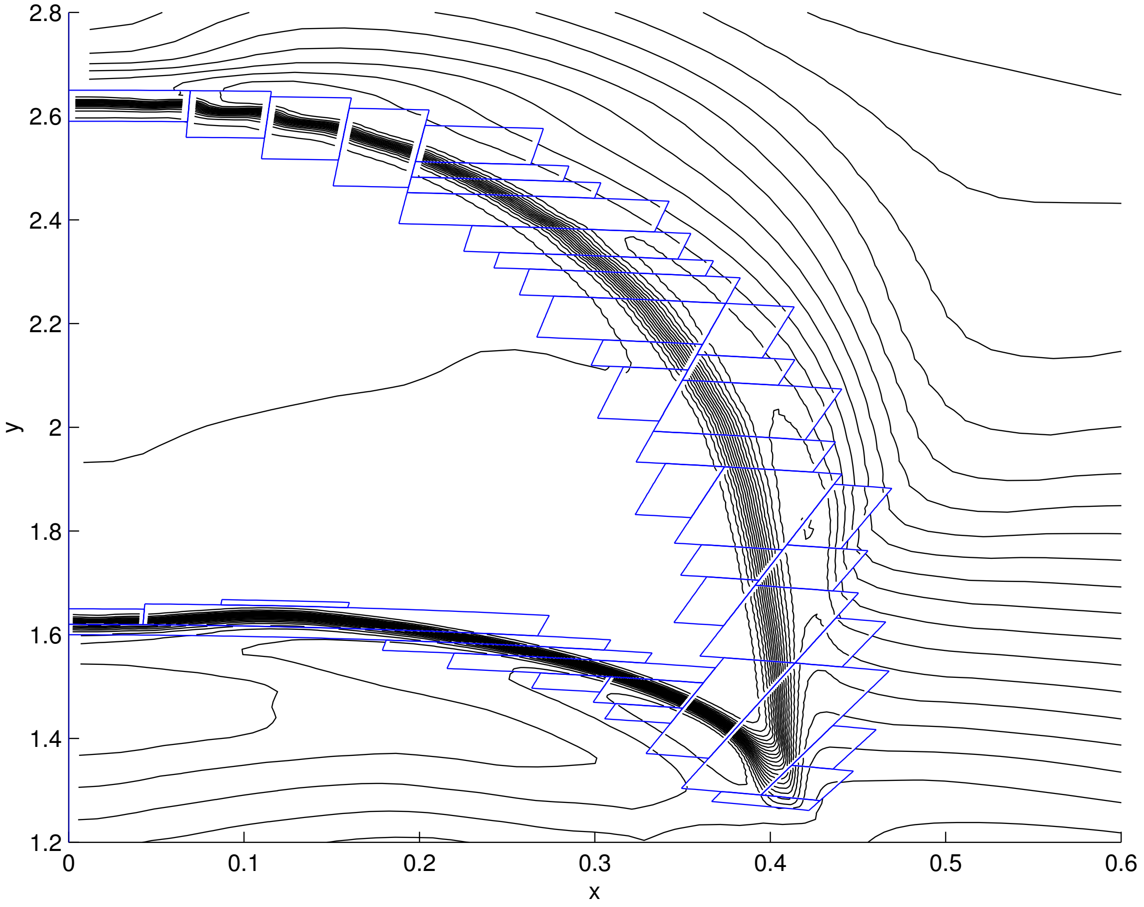

Fig. 7.19.— The "hot bubble" problem solved with magnetic field $b = 0.4$. 50 density contour between 0 and 1 are used.

REFERENCES

D. S. Balsara, Total variation diminishing scheme for adiabatic and isothermal magnetohydrodynamics, *Apjs*, 116 (1998), 133-153.

M. J. Berger and P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* 82 (1989), 64-84.

M. J. Berger and J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.* 53 (1984), 484-512.

M. J. Berger and I. Rigoutsos, An algorithm for point clustering and grid generation, *IEEE Trans. on Systems, Man, and Cybernetics*, 21 (1991).

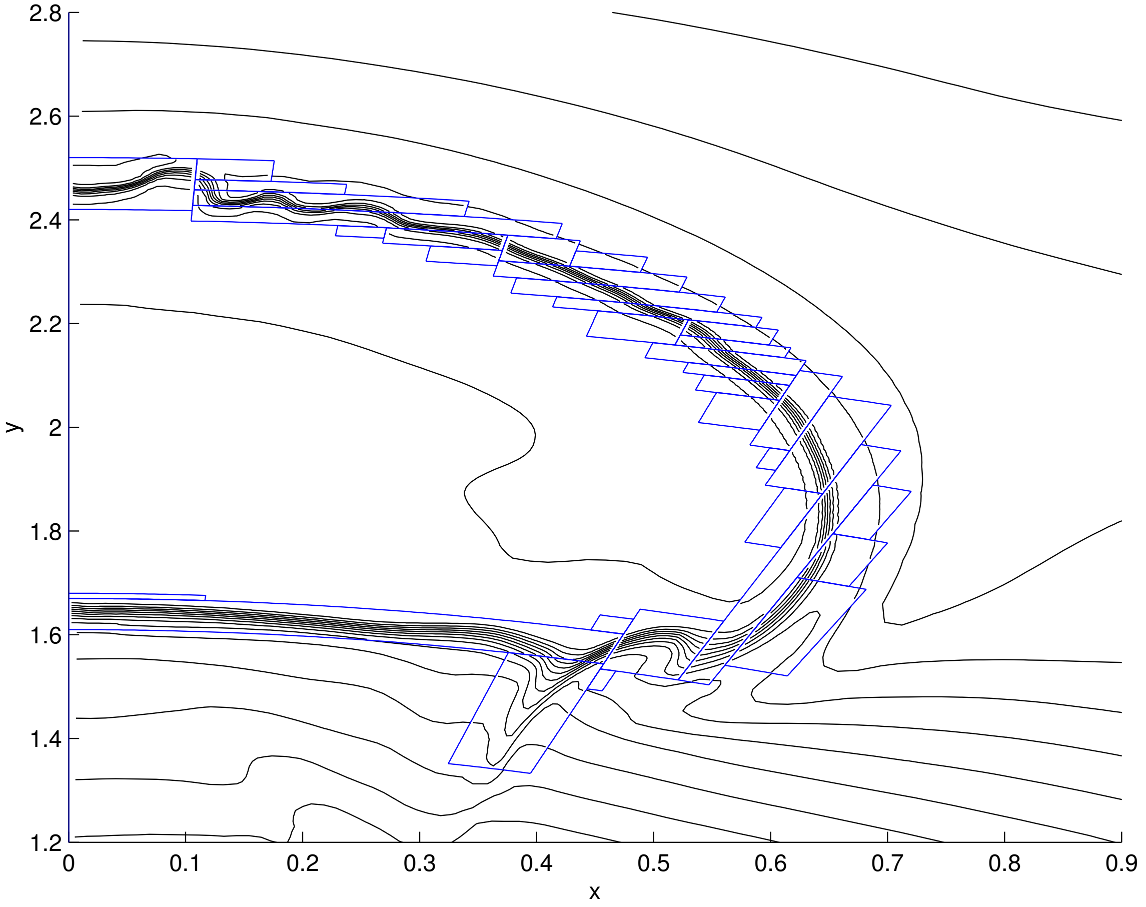

Fig. 7.20.— The "hot bubble" problem solved with magnetic field $b = 0.2$. 50 density contour between 0 and 1 are used.

D. S. Balsara and D. S. Spicer, A staggered mesh algorithm using high order Godunov fluxes to ensure solenoidal magnetic fields in magnetiohydrodynamics simulations, *J. Comput. Phys.* 149 (1999), 270-292.

D. S. Balsara, Divergence-free adaptive mesh refinement for magnetohydrodynamics, *J. Comput. Phys.*, 174 (2001), 614-648.

M. Brio and C. C. Wu, An upwinding differencing scheme for the equations of ideal magnetohydrodynamic. *J. Comput. Phys.*, 75(1988), 400.

P. Batten, N. Clarke, C. Lambert, and D. Causon, On the Choice of Wave Speeds for the HLLC Riemann solver. *SIAM J. Sci. Comput.*, 18 (1997), 1553-1570.

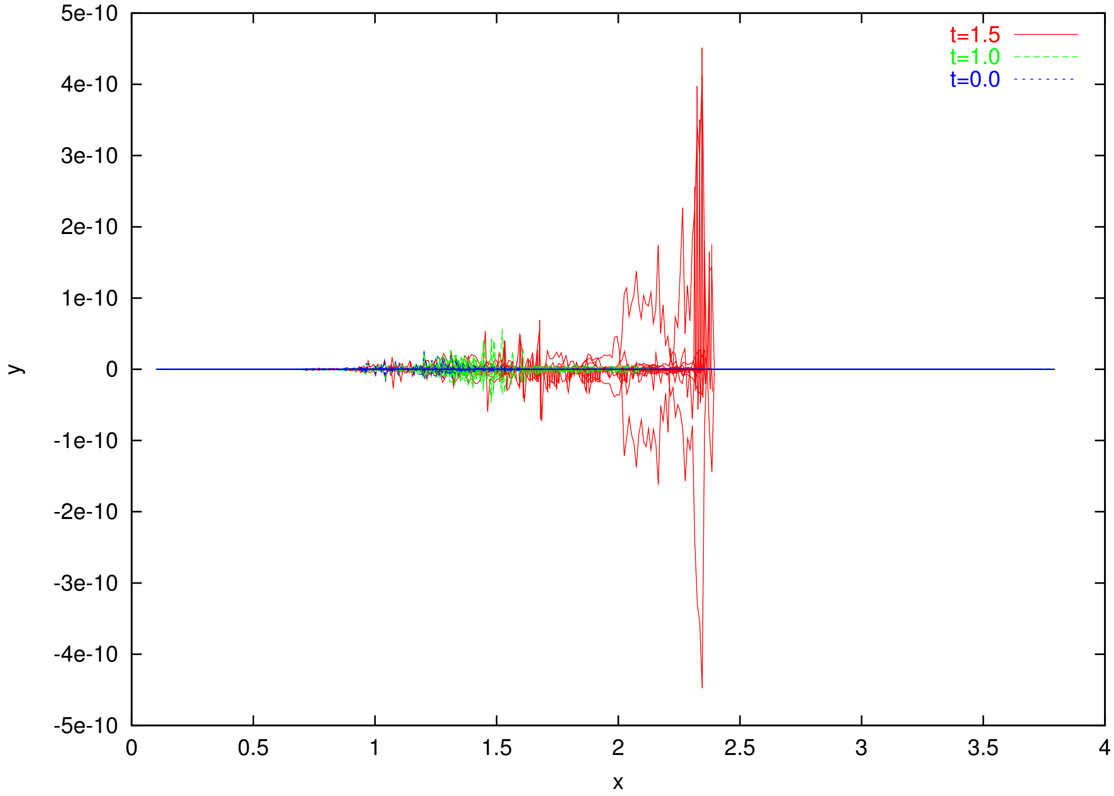

Fig. 7.21.— The $\nabla \cdot B$ of the "hot bubble" problem for the adaptive mesh refinement with refinement ratio of 3. $t = 1.5$

P. Colella, Dan Martin, and other staffs in applied numerical analysis group, Chombo1.2, released in 2002. http://seesar.lbl.gov/anag/chombo/index.html. Lawrence Berkley Nat'l Lab.

D. A. Clarke, A consistent method of characteristics for multidimensional magnetohydrodynamics, *ApsJ*, 457 (1996), 291-320.

R.J. Leveque, http://www.amath.washington.edu/ claw/, 2003.

W. Dai and P. R. Woodward, A simple finite difference scheme for multidimensional magnetohydrodynamics, *J. Comput. Phys.* 142 (1998), 331-369.

B. Einfeldt, On Godunov-type methods for gas dynamics, *SIAM J. Numer. Anal.*, 25 (1988), 294.

B. Einfeldt, C. D. Munz, P. L. Roe, and B. Sjogreen, On Godunov-type methods near low densities, *J. Comput. Phys.*, 92 (1991), 273.

B. Fryxell, K. Olson, P. Ricker, F. X. Timmes, M. Zingale, D. Q. Lamb, P. MacNeice, R. Rosner, J. W. Truran, H. Tufo, Flash: An adaptive mesh hydrodynamics code fro modeling astrophysical thermonuclear flashes. *ApJS*, 131 (2000), 273.

A. Harten, P. D. Lax, and B. van Leer, On upstreaming differencing and Godunov-type schemes for hyperbolic conservation laws, *SIAM Rev.*, 25 (1983), 35.

S. Li and Mac Hyman, Solution adapted nested grid refinement for 2-D PDEs, 1998, Technical Report, LA-UR-98-5463, Los Alamos National Lab.

S. Li and Mac Hyman, Interactive and dynamic control of adaptive mesh refinement with nested hierarchical grids, 1998, Technical Report, LA-UR-98-5462, Los Alamos National Lab.

S. Li, J. M. Hyman, L. Petzold, Solution Adapted Nested Grid Refinement and Sensitivity Analysis for Parabolic Partial Differential Equations, Proceedings of the First Sandia Workshop on Large-Scale PDE-Constrained Optimization, Springer's Lecture Notes in Computational Science and Engineering, O. Ghattas, editor. Santa Fe, New Mexico, USA, 4-6 April 2001. in Springer CSE book series.

G. S. Jiang and C.-C. Wu, A highg-order WENO finite difference scheme for the equations of ideal Magnetohydrodynamics, *J. Comput. Phys.*, 150 (1999) 561-594.

G. S. Jiang and C. W. Shu, efficient implementation of weighted ENO schemes, *J. Comt. Phys.* 126 (1996), 202.

P. Janhunen, A positive conservative method for magnetohydrodynamics based on HLL and Roe methods, *J. Comput. Phys.* 160 (2000), 649.

B.-I. Jun and M. L. Norman, A numerical study of Rayleigh-Taylor instability in magnetic fluids, *ApsJ*, 453 (1995), 332-349.

A. Kurganov and D. Levy, A third order semi-discrete central scheme for conservation laws and convection-diffusion equations, *SIAM J. Sci. Comput.*, 22 (2000), 1461-1488.

B. Van Leer, Towards the ultimate conservative difference scheme II, monotonicity and conservation combined in a second order scheme, *J. Comput. Phys.* 14(1979), 361-470.

D. Levy, G. Puppo and G. Russo, A third order central WENO scheme for 2D conservation laws, *Appl. Numer. Math.* 33 (2000), 407-414.

D. Levy, G. Puppo and G. Russo, Compact central WENO schemes for multidimensional conservation laws, 2002, preprint.

R.J. Leveque, Intermediate boundary conditions for time-split methods applied to hyperbolic partial differntial equations, *Math. Comp*, 47 (1986), 37-54.

S. Li, Adaptive mesh methods and software for time-dependent partial differential equations, Technical Report TR98-038, Dept. of Computer Science, University of Minnesota, 1998.

S. Li and H. Li, A novel approach of maintaining divergence-free for adaptive mesh refinement, submitted to JCP.

S. Li and H. Li, A HLLC Riemann solver for MHD, preprint, 2003

S. Li and J. M. Hyman, Local time step versus locked time step, Technical report, 2002, Los Alamos Nationa Lab.

E. F. Toro, M. Spruce, and W. Speares, Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4(1994), 25-34.

P. MacNeice, K. M. Olson, C. Mobarry, R. Fainchtein, and C. Packer, PARAMESH: A parallel adaptive mesh refinement community toolkit., *Comput. Phys. Comm.*, 126 (2000), 330-354.

S. Mitra, M. Parashar, J. C. Browne, Dept. of Computer Science, Univ. of Texas at Austin, 1997; at http://www.caip.rutgers.edu/ parashar/DAGH/.

R.B. Pember, J.A. Greenough, and P. Colella, An Adaptive Higher-Order Godunov Method for Gas Dynamics in Three Dimensional Orthogonal Curvilinear Coordinates, Lawrence Livermore National Laboratory technical report UCRL-JC-123351 (1996).

K. G. Powell, P. L. Roe, T. J. Linde, T. I. Gombosi, and D. L. DeZeeuw, A solution-adaptive upwinding scheme for ideal magnetohydrodynamics, *J. Comt. Phys.*, 154 (1999), 284.

J.-F. Remacle, J. E. Flaherty, and M. S. Shephard, An adaptive discontinuous Galekin technique with an orthogonal basis applied to compressible flow problems. *SIAM Review*, 45 (2003), 53-72.

Dongsu Ryu and T. W. Jones, Numerical magnetiohydrodynamics in astrophysics: algorithm and tests for one-dimensional flow., *The astrophy. Journal,* 442 (1995), 228-258.

Dongsu Ryu, Magnetohydrodynamics code basics, *J. Korean Astro. Society*, 34 (2001), 209-213.

P. L. Roe and D. S. Balsara, Notes on the eigensystem of magnetohydrodynamics, *SIAM J. Appl. Math.*, 56 (1996) 57-67.

C. W. Shu and S. J. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes II, *J. Comput. Phys.*, 83(1989), 32-78.

G. Strang, On the construction and comparison of different schemes, *SIAM J. Numer. Ana.*, 5 (1968), 506-517.

J. M. Stone and M. L. Norman, ZEUS-2D: A radiation magnetohydrodynamics code for astrophysical flows in two space dimensions ii. The magnetohydrodynamics algorithms and tests, *Astrophys. J. Suppl.* 80 (1992), 791.

Gabor Tóth, The $\nabla \cdot B = 0$ constraint in shock-capturing magnetohydrodynamics codes, *J. Comt. Phys.*, 161 (2000), 605-652.

K. S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equation in an isotropic media, *IEEE Trans. Antenna Propagation* AP-14 (1966), 302.

M. Wesenberg, Efficent MHD Riemann solvers for simulations on unstructured triangular grids, *J. Numer. Math.*, 10 (2002), 37-71.

M. Zingle, et al. Mapping initial hydrostatic models in Godunov codes, *ApsJ, Sup*, 143 (2002), 539-565.

This preprint was prepared with the AAS LaTeX macros v5.0.